SYNTACTIC ANALYSIS OF THE RUSSIAN SENTENCE

[illegible] J. Plath
Aleksandr Andreyewsky
Robert E. Shorr, et al
The IBM Corporation
Thomas J. Watson Research Center

TECHNICAL REPORT NO. RADC-TR-67- 64
October 1967

This document is subject to special
export controls and each transmittal
to foreign governments, foreign na-
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

Rome Air Development Center
[illegible]
Griffiss Air Force Base, [illegible]

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded, by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

SYNTACTIC ANALYSIS OF THE RUSSIAN SENTENCE

Dr. Warren J. Plath
Alexander Andreyewsky
Robert E. Strom, et al
The IBM Corporation
Thomas J. Watson Research Center

This document is subject to special export controls and each transmittal to foreign governments, foreign nationals or representatives thereto may be made only with prior approval of RADC (EMII), GAFB, N.Y. 13440. The distribution of this document is limited because it is embargoed under the U.S. Export Control Act of 1949.

FOREWORD

This final technical report summarizes the research performed during the period May 1965 to May 1967 at the IBM Thomas J. Watson Research Center at Yorktown Heights, N.Y. 10598 under Contract AF30(602)-3782, Project 4599. The authors of this report were Dr. Warren J. Plath and Messrs Alexander Andreyewsky, Robert E. Strom and Erhard O. Lippman. The project engineer was Zbigniew L. Pankowicz, Rome Air Development Center, EMIIH, Griffiss Air Force Base, N.Y. 13440.

Information in this report is embargoed under the U.S. Export Control Act of 1949 administered by the Department of Commerce. This report may be released by departments or agencies of the U.S. Government to departments or agencies of foreign governments with which the United States has defense treaty commitments. Private individuals or firms must comply with Department of Commerce export control regulations.

This technical report has been reviewed and is approved.

Approved:
FRANK J. TOMAINI
Chief, Intel & Info Processing Branch
Intel & Info Processing Div.

Approved: fr Howard Davis
JAMES J. DIMEL, Colonel, USAF
Chief, Intel & Info Processing Div.

FOR THE COMMANDER Irving J. Gabelman
IRVING J. GABELMAN
Chief, Advanced Studies Group

ABSTRACT

This report summarizes research performed at the IBM Thomas J. Watson Research Center during the past two years in the area of automatic syntactic analysis of the Russian sentence. The activities described and the results presented relate to two distinct surface structure parsing systems for Russian. The main emphasis of the project was on the design and development of the Combinatorial Syntactic Analysis (CSA) system, accompanied by an extensive program of linguistic research on Russian grammar. A considerably smaller effort, conducted in parallel with that on CSA, was concerned with further work on multiple-path predictive syntactic analysis of Russian.

The Combinatorial Syntactic Analysis system is an exhaustive automatic sentence parsing system which produces surface structure descriptions of sentences by systematically forming all grammatical combinations of adjacent pairs of constituents in a bottom-to-top, left-to-right sequence. The grammars and syntactic coding accepted by the system are written in a special metalanguage in which grammatical constituents are treated as structured symbols consisting of a part-of-speech name followed by a string of tags, or attribute/value pairs. Because of the extensive facilities provided by the metalanguage for introducing tags and defining operations on them, grammars developed for the CSA system can be made significantly more powerful and compact than those of the conventional IC type.

Part I of this report describes the parsing algorithm, the tag metalanguage, and the overall organization of the Combinatorial Syntactic Analysis system. More detailed accounts of the logical organization of the CSA parser and the associated Dictionary Assembly/Update system are included in two appendices to that section. A description of linguistic research on the CSA Russian grammar is presented in Part II, along with a brief summary of related language processing activities. In addition to a presentation of representative rules of the experimental Russian grammar developed and a report on subclassification studies, Part II includes an extensive account of further investigations of such key topics in Russian grammar as apposition, predication, and coordination

Part III, the final section of the report, summarizes research activities on predictive syntactic analysis of Russian. The main accomplishments in this area were 1) bringing the multiple-path predictive Russian Syntactic Analyzer into operational status at IBM Research, 2) expansion of the dictionary, and 3) terting and evaluation of the performance of the Analyzer on several thousand words of Russian text.

[redacted]

TABLE OF CONTENTS

I. THE CSA SYSTEM: ANALYSIS ALGORITHM, METALANGUAGE, AND SYSTEM ORGANIZATION

II. THE CSA RUSSIAN GRAMMAR: LINGUISTIC RESEARCH AND RELATED LANGUAGE PROCESSING ACTIVITIES

page

FVALUATION

1. Subject R&D effort constitutes a transition rrom the limited environment syntactic analysis, embodied in the bidirectional single pass technique of the Mark II system, to the sentence wide syntactic analysis for an ultimate adaptation to a software system for production of syntactic level Russian-to-English machine translations of scientific and technical texts.

2. The main effort was directed toward development of the combinatorial syntactic analysis system for exhaustive parsin of Russian sentences. It was combined with a thorough lingui research on the Russian grammar for combinatorial syntactic analysis. Subclassification of parts of speech according to their syntactic-semantic features deserves a special attentio in this effort. Equally significant is the adoption of the linguistic notion of "slovosochetaniye" (grammatically bound word group) for a further sophistication of the syntactic recognition program.

3. A small scale research effort in the predictive syntactic analysis was conducted in parallel with the work on the combinatorial syntactic analysis system, in order to compare the relative merits and deficiencies of both these surfact struc parsing systems. This effort was aimed primarily at bringing the Russian Predictive Syntactic Analyzer into the operational status. The Russian-English structural transfer study, intenc in this effort, did not yield any significant results due to the fact that the output from the Analyzer was not available until the end of the contract period. Research on the predictive syntactic analysis has revealed the desirability of transformational syntactic recognition. The report is concluded with the following meaningful statement: While the prospect of constructing a huge array of interlocking "microgrammars" in order to handle texts of various types is an extremely uninviting one, the possibility of constructing a more restrictive grammar adequate for a single specific field appears worthy of serious exploration."

ZBIGNIEW L. PANKOWICZ
Technical Evaluator

I. THE CSA SYSTEM: ANALYSIS ALGORITHM, METALANGUAGE, AND SYSTEM ORGANIZATION

Erhard O. Lippman
Warren J. Plath
Robert E. Strom

I. THE CSA SYSTEM: ANALYSIS ALGORITHM, METALANGUAGE, AND SYSTEM ORGANIZATION

1.0 Introduction

The Combinatorial Syntactic Analysis (CSA) system is an exhaustive sentence parsing system which has been implemented in the FAP language on the IBM 7094. When supplied with a set of grammar rules, together with appropriate syntactic alternatives for each word in a sentence, the CSA system produces all surface structure analyses of the sentence that are consistent with the rules of the grammar and the syntactic coding of the words. The analysis algorithm builds up structural descriptions of a sentence by systematically forming all grammatical combinations of adjacent constituent pairs in a bottom-to-top, left-to-right sequence. When the process terminates, following formation of the final combination involving the last alternative of the last word in the sentence, both complete sentence structure trees and various intermediate results are retrieved, edited, and printed out.

The grammars and syntactic coding accepted by the CSA system are written in a special metalanguage in which grammatical constituents are treated as structured symbols consisting of a constituent name followed by a string of <u>tags</u>, or attribute/value pairs. In their overall form, grammar rules are currently limited to those of the binary immediate constituent (IC) type; that is, all rules are of the general form $C_1 + C_2 = C_3$, which signifies that a constituent of type C_1 can be combined with an immediately following constituent of type C_2 to form a constitute (or higher-order constituent) of type C_3. Because of the extensive facilities provided by the metalanguage for introducing tags and defining operations on them, however, grammar rules employed in the CSA system can be made significantly more powerful and compact than those of conventional IC grammars.

In the work on syntactic analysis of Russian, the use of tags has been especially valuable in dealing effectively with a variety of syntactic properties of Russian constructions, in particular, agreement and government relationships involving such attributes as case, number, and gender. Moreover, since it permits free introduction of tags in both grammar rules and dictionary coding without changing the analysis program, the metalanguage provides a convenient vehicle for experimental investigation of new syntactic and semantic relationships. The flexibility of the tag notation has also been underscored by the complete ease with which token grammars of other languates (English, German, and Hungarian) have been accepted and applied by the CSA system. Before presenting detailed descriptions of the metalanguage (Section 1.2) and of the CSA system organization (Section 1.3), a brief sketch will be given of the analysis algorithm employed by the CSA system.

1.1 The Analysis Algorithm

Beneath an overlay of tag operations, which will be discussed below in Section 1.2, the analysis algorithm employed by the CSA system makes use of a parsing strategy which is similar to one originally described by Sakai (1962), but whose specific details are due to Kuno (1965). The general flow of the CSA parsing algorithm is as follows.

Before actual parsing begins, each word in the text is supplied with syntactic alternatives by a process of dictionary lookup. These alternatives represent various mutually exclusive syntactic properties of a word form: for example, PEC6 can be a noun ('oven') or an infinitive form of a verb ('to bake'), STOL ('table') can be a masculine singular noun in either the nominative or accusative case, and so on.

The analysis algorithm treats each sentence as a unit, processing the syntactic alternatives of its component words from left to right. The alternatives are read one at a time into a work area which occupies a large storage matrix. As a given alternative is read in, it is assigned boundary markers indicating what word it spans in the sentence; for example, all syntactic alternatives of the third word in a sentence are assigned the boundary markers (3, 3). The syntactic alternative and its associated markers are then entered in the first available row of the storage matrix.

At this point, using the boundary markers as a guide, the parser pairs the new syntactic alternative with each entry in the matrix which represents a constituent that is left-adjacent to it in the sentence. As each pair is formed, it is looked up in the table of grammar rules to determine whether or not its components can legitimately be combined into a constitute, or higher-order constituent. If they can, a new row is created in the matrix for each valid combination. Each such row contains not only a pair of boundary markers, indicating the part of the sentence spanned by the higher-order constituent, but also a pair of numbers indicating the locations of its immediate constituents in the matrix.

When all combinations of a syntactic alternative with its left-adjacent neighbors have been tried, the algorithm moves on to the following rows of the matrix and searches for additional combinations by successively pairing each of the new constitutes with each of its left-adjacent neighbors. As soon as no further rows remain to be processed, a new syntactic alternative is read in, and the above process is repeated. The process terminates when the program attempts to read in another alternative at a point where none remains to be processed.

As a simple illustration of the operation of the parsing algorithm, consider the case of a hypothetical sentence whose five component words have been assigned the (unique) string of syntactic alternatives displayed in (1) and which is to be parsed exhaustively using the grammar rules (2).

(1) A N V N N
1 2 3 4 5

(2) A + N = N
N + V = S
N + N = N
V + N = V

The course of the analysis process can be followed by examining successive rows of Table I-1, which displays the final contents of the storage matrix for the grammar and sentence under consideration. The process begins when the first syntactic alternative (A) is read into the first row of the previously empty storage matrix. Since there are no entries in the matrix that represent left-adjacent sentence neighbors with which A can be paired, the next syntactic alternative (N) is immediately read into row 2. The program then searches for all matrix entries whose rightmost word number is one less than that of the leftmost word spanned by the current item, since this is precisely the condition for left-adjacency. The entry in line 1 fulfills the condition; hence, it is paired with the entry in line 2, forming the couple (A, N), which is looked up in the table of grammar rules.

The (A, N) pair matches the left half of a rule in the grammar, indicating that (A, N) represents a grammatically permissible combination. The right half of the same rule indicates that the resultant constituent is an N. The analysis program copies the result in the third row of the matrix, along with indications (a) that the new constituent spans words 1 and 2 in the sentence and (b) that its components occupy rows 1 and 2 of the matrix.* Since all combinations of the entry in row 2 with left-adjacent neighbors have now been exhausted, the program proceeds to row 3. When no candidates are found for combination with that entry (at that point, the last one in the matrix), the program reads the next syntactic alternative (V) into row 4. The parsing process continues in this fashion until all possibilities have been exhausted, yielding the storage matrix configuration of Table I-1.

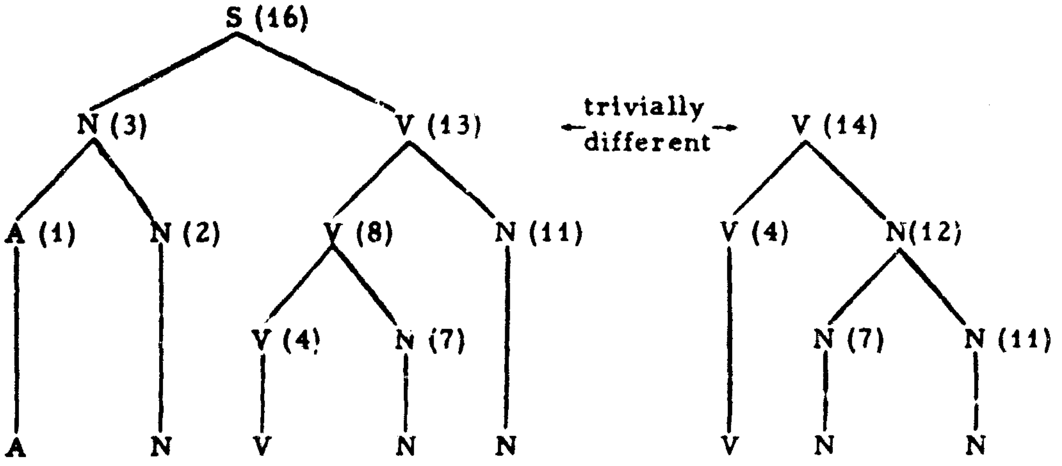
*It should be noted that, in dealing with actual grammars, there may be several grammatically acceptable ways of combining a given ordered pair of constituents. Such alternatives are represented by a collection of subrules grouped under a heading consisting of the constituent pair in question. Whenever a constituent pair in the sentence matches the heading of such a rule, it is tested against all subrules of that rule, and each one that applies gives rise to a new row in the storage matrix.

Table I-1. Storage Matrix for Combinatorial Syntactic Analysis of a Sample Sentence

	Boundary Markers of String Spanned		Row Numbers of Subconstituents		
Row Number	Leftmost Word	Rightmost Word	Left Constituent	Right Constituent	Name of Constituent
1	1	1	-	-	A
2	2	2	-	-	N
3	1	2	1	2	N
4	3	3	-	-	V
5	2	3	2	4	S
6	1	3	3	4	S
7	4	4	-	-	N
8	3	4	4	7	V
9	2	4	2	8	S
10	1	4	3	8	S
11	5	5	-	-	N
12	4	5	7	11	N
13	3	5	8	11	V
14	3	5	4	12	V*
15	2	5	2	13	S
16	1	5	3	13	S

*Since its behavior will duplicate that of the constituent in row 13, the constituent in row 14 is prevented from entering into further combinations.

In processing the contents of the matrix prior to final printout, the program finds only one "complete" analysis, corresponding to the S in row 16, which spans the entire sentence. In addition, there is a partial analysis (corresponding to the V in row 14), the remainder of which has been suppressed because it will duplicate corresponding portions of the first analysis. In tree format, with row numbers indicated in parentheses opposite each node, the analyses look as follows:



1.2 The Tag Metalanguage

Much of the power and flexibility of the CSA system as a research tool is attributable to properties of the metalanguage in which the grammar rules are written. As has already been noted above, a key feature of the metalanguage is the treatment of grammatical constituents as structured symbols, each of which consists of a constituent name followed by a string of tags, or attribute/value pairs. Although tags are not restricted to binary values, strings of tags have obvious formal similarities to the syntactic feature vectors employed in recent formulations of transformational grammar (Chomsky, 1965). However, the principal influences on the development of the tag metalanguage have been two earlier systems that have employed structured symbols: the COMIT programming language (Yngve, 1961), with its logical subscripts, and the grammatical index notation employed in multiple-path predictive analysis of Russian (Plath, 1963).

The present metalanguage shares with the grammatical index notation the property of being a rule-writing language in which variables play an important role, but it is also endowed with a COMIT-like facility for ad-lib introduction of names of constituents, attributes, and values. The following is a detailed formal description of the properties of the metalanguage.

1.2.1 Rule and Subrule Format

As was noted in Section 1.1, the analysis program systematically tests each pair of adjacent constituents with part-of-speech codes C_1 and C_2 against all subrules grouped under the corresponding heading. Accordingly, it is necessary in preparing a grammar for the system to organize the rules into "packets", each containing the complete set of subrules for a given ordered part-of-speech pair. Every subrule consists of the following parts:

[(⟨label⟩)] ⟨part of speech⟩⟨tag conditions⟩ + ⟨part of speech⟩⟨tag conditions⟩ = ⟨part of speech⟩⟨tag replacements⟩[⟨transfer section⟩]

The above notation signifies: A subrule contains an optional label, enclosed in parentheses, followed by a part of speech (C_1), tag conditions, a plus sign, a part of speech (C_2), tag conditions, an equals sign, a part of speech (C_3), tag replacements, and optionally a transfer section. The section of the subrule between the label and the equals sign is called the left half of the subrule; the part between the equals sign and the transfer section is called the right half of the subrule. As we have seen, the left half of a subrule contains a pair of parts of speech with tag conditions, and the right half contains a part of speech with tag replacements. All subrules with the same ordered pair of part-of-speech codes in their left half must be grouped together, and preceded by a rule header, written:

*⟨part of speech⟩ + ⟨part of speech⟩

A rule header, followed by one or more subrules having the same part-of-speech pair in the left half as was written on the rule header, is called a rule packet. The following is a sample rule packet taken from a simplified grammar of Cerman:

```
*PRON + VERB
PRON SC/PERS TY/E CS/NOM NR/X + VERB SC/FIN TY/E
NR/X = PRED SC/PERS SS/PRVAUX NR/X
PRON CS/X + VERB G1/X = VERB TY/P G1/2-X ETC/2
```

In the rule packet just illustrated, the part-of-speech pair is (PRON, VERB) and this pair appears as the two parts of speech in the left-half sections of the illustrated subrules. These subrules do not happen to contain labels or transfer sections. However, each has a different set of tag conditions, and a different tag replacement section. Whenever two adjacent constituents are encountered in the course of parsing a sentence, and the left constituent has part of speech PRON, while the right constituent has part of speech VERB, the rule packet for PRON + VERB will be executed. Any subrules whose tag conditions are met will cause a new constituent to be generated whose part of speech is given by that of the right half, and whose tags are determined by the pattern given in the tag replacement section. For example, if the two constituents to be combined are:

PRON SC/PERS TY/E CS/NOM NR/SING and
VERB TENSE/PRES NR/SING SC/FIN TY/E G1/DAT, ACC REFL/NO

then the first subrule in the rule packet shown above will be successful. The "variable" X will be set to "SING" and the new constituent dominating the above two constituents will be:

PRED SC/PERS SS/PRVAUX NR/SING.

If the PRON given above were marked TY/G or NR/PLUR, then the tag conditions would not be met and no new constituent would be written.

If the two constituents to be combined are:

PRON SC/PERS TY/E CS/DAT NR/PLUR and
VERB SC/INF TY/E NR/NONE TENSE/NONE G1/ACC, DAT

then the second subrule will "succeed" and the following new constituent will be created:

VERB TY/P G1/ACC NR/NONE TENSE/NONE SC/INF.

If now, immediately to the left of the first PRON, there is another PRON with PRON SC/PERS TY/E CS/ACC NR/SING, then this constituent will be adjacent to the new constituent spanning both the first PRON and VERB, and can combine with it according to the same rule, producing:

VERB TY/P G1/   NR/NONE TENSE/NONE SC/INF.

1.2.2 Data Tags and Data Constituents

During the course of analysis, any two adjacent constituents are tested against the rule packet, if any, for the ordered part-of-speech pair which they constitute. If there are successful subrules in this rule packet, new constituents spanning the range of word numbers spanned by both original constituents are created. The initial constituents, which must be present before any constituents can be generated by application of a subrule, are the syntactic alternatives assigned by dictionary lookup to each of the words of the sentence. These constituents normally have no "subconstituents" (i.e., pairs of constituents from which a constituent is generated by application of a rule and which are the subnodes of the constituent in the generated tree) and are of unitary word-span. The constituents which are matched against the rules during parsing, whether they come from the original input data or whether they have been created by previous applications of grammar rules, are generally called data constituents and consist of a part-of-speech name plus a number of tags, which, in order to distinguish them from the similarly-written components of rules called tag conditions, are sometimes called data tags. Hence the tag metalanguage in which rules are written is a language containing tag conditions and tag replacements for matching and generating data tags.

The form of data constituents

Data constituents have the following (externally printed or punched) form: (part of speech)(tags).

The part of speech has the form: (symbol), where a symbol is any string of six or fewer consecutive alphanumeric characters, the first of which is alphabetic. The alphabetic characters are:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z *
The numeric characters are:
0 1 2 3 4 5 6 7 8 9
The following are legal symbols:
PRON ABC1 G3*Q
The following are not legal symbols:
3A (first character not alphabetic)
AB C (characters separated by a blank)
PRONOUN (more than six characters)
AB(C (contains non-alphanumeric character)

The tags field is a list of zero or more individual tags, separated by blanks, Each tag has the following form: (attribute)/(value), where a value has the form: * or $ or (constant symbol)
or (constant symbol), ... (constant symbol)
where an attribute is any symbol, and a constant symbol is any symbol beginning with a letter from A through W.

The following are valid data constituents:
ART
PRON CS/NOM
VERB G1/* G2/ACC, DAT TENSE/PRES NUMB/PLUR PERSON/$
NOUN TY/PHRASE LEFT/NO RIGHT/SI CASE/ADESS.

The following are not valid data constituents:
ART/DEF (part of speech must be present)
ART DEF/YES (a constant must not begin with X, Y, Z, or *)
NOUN CASE/S1, $ (the dollar sign must stand alone)
VERB TYP/E, TENSE/PRES (the tags must be separated by blanks, not commas)
NOUN G1/ CASE/ALLAT (some value must be given -- the * has the semantic meaning of a null value and should

be punched; it will not be printed)
NOUN PERSON/3 NUMB/SING (a numeral is not a symbol as defined above)

1.2.3 Subrule Tags and Subrule Constituents

As was mentioned earlier, rules are organized into rule packets, consisting of one or more *subrules* with a common part-of-speech pair. Furthermore, each subrule consists of an optional label, a left half, an equals sign, a right half, and an optional transfer section. The left half contains a part-of-speech code with associated tag conditions for each of the two constituents combined by the subrule.

The form of tag conditions (left half)

Tag conditions have the following form:

```
     ⟨tag conditions⟩ = ⟨empty⟩
or   ⟨tag condition⟩ [... ⟨tag condition⟩]
     ⟨tag condition⟩ = ⟨test attribute⟩/⟨specifier⟩
     ⟨specifier⟩ = *
or   ⟨constant symbol⟩[, ⟨constant symbol⟩ ...]
or   ⟨variable⟩
or   ⟨variable⟩ - ⟨constant symbol⟩[, ⟨constant symbol⟩...]
or   (⟨constant symbol⟩)
or   (⟨variable⟩)
     ⟨variable⟩ = any symbol beginning with X, Y, or Z.
```

When a subrule is tested, it is known that the two data constituents have the part-of-speech codes specified in the subrule, for only that rule packet headed by the appropriate part of-speech pair is executed when the two adjacent constituents are matched against the grammar. The tag conditions on the subrule are executed sequentially, and each may either *succeed* or *fail*. When a tag condition succeeds, the next tag condition is tried, until *all* tag conditions have been tested successfully, in which case the subrule is said to *succeed*. When a tag condition fails, no more tag conditions are tried and the subrule is said to *fail*. In addition to causing a subrule to succeed or fail, tag conditions can have side effects. The principal side effect of a tag condition is variable-setting, since this affects a future tag condition or replacement within the subrule. The following are the different types of tag conditions tabulated according to the type of specifier which can appear in them.

Tag conditions classified according to specifier type

1. *: If the test attribute is not present on the data constituent being tested, or if the attribute is present but has no value, then the tag condition

succeeds. If the attribute is present with some value, the tag condition fails.

2. ⟨constant symbol⟩: If the test attribute is present on the data constituent and the constant symbol appears in its value field, then the tag condition succeeds; otherwise, it fails.

3. ⟨constant symbol⟩,... ⟨constant symbol⟩: If the test attribute is present on the data constituent and at least one of the constant symbols in the list appears in its value field, the tag condition succeeds; otherwise, it fails.

4. ⟨variable⟩: If the test attribute is not present on the data constituent, the tag condition fails. Otherwise, there are two cases:

   Case a: If this is the first occurrence of the variable in the subrule, then the variable must be defined. This is accomplished by assigning the variable the entire contents of the value field of the test attribute on the data constituent. (Such an assignment may be modified during further processing of the current pair of data constituents in accordance with the subrule in question. In any event, it does not remain in force after completion of the processing of the current constituent pair according to that subrule.)

   Case b: If this is not the first occurrence of the variable in the subrule, it has previously been defined during this application of the subrule to the current data constituent pair. Delete from the defined value of the variable all items which do not appear in the value field of the test attribute on the data item. If nothing remains, the tag condition fails. If one or more values remain, then these are precisely the values common to both the data constituent attribute from which the variable was defined and the data constituent attribute currently being tested, and the tag condition succeeds.

5. ⟨variable⟩ - ⟨constant symbol⟩[, ⟨constant symbol⟩...]: First, as in 4, either the variable is defined, or the already defined variable is tested. In addition, all constants appearing on the list after the minus sign in the rule are also deleted from the definition of the variable. If nothing remains, the tag condition fails. If one or more values remain, they are guaranteed (in the absence of repetitions) to be other than those in the list of constants after the minus sign (the "exclusion list"), and the tag condition succeeds.

6. (⟨constant symbol⟩): The same as 2 above, provided that the test attribute is present on the data constituent. Unlike 2, however, if the test attribute is not present, the tag condition does not fail, and execution of the subrule continues.

7. (⟨variable⟩): The same as 4 above, except that the tag condition cannot fail unless the test attribute is present on the data constituent. If the attribute does not appear, the variable is defined with null value, but execution of the subrule continues.

Note: When the $ appears on a data constituent as the value of an attribute, it matches any constant from a tag condition of type (2) above, as well as any constant referred to by a variable in a tag condition of type (4). When * appears as the value of an attribute on a data constituent, it represents a null value and hence can only satisfy a tag condition of type (1).

Right half of a subrule

When a subrule is applied, its tag conditions are executed interpretively as instructions to make tests and to set variables. If the subrule fails, no new constituent is created. If the subrule succeeds, a new constituent is created according to the specifications of the part of speech and tag replacement section of the right half of the subrule. The part of speech in the right half of a rule can be either a constant symbol (in which case that constant becomes the part of speech of the rewritten constituent), or a variable (in which case the value defined for that variable -- which must be restricted to a single value -- becomes the part of speech of the rewritten constituent). After the part of speech come the tag replacements, each of which has the form:

⟨attribute⟩/⟨replacement specifier⟩
or ETC/1 or ETC/2 or ETC/1, 2

The attribute must be a constant symbol; the permissible tag replacement specifiers with their interpretations are as follows.

Tag replacements classified according to specifier type

1. ⟨constant symbol⟩ or ⟨constant symbol⟩, . . . ⟨constant symbol⟩. The attribute, along with the constant symbol(s) specified in the subrule, is added to the tags field of the constituent being generated.

2. *: The attribute, with no value, is added to the tags field of the constituent being created.

3. ⟨variable⟩[, ⟨constant symbol⟩, . . .]: The attribute with a value consisting of the defined value of the variable plus the constant symbols, if any, is added to the tags field of the generated constituent.

4. ⟨numeral⟩ - ⟨variable⟩: The numeral is either 1 or 2, referring to the left subconstituent (C_1) or the right subconstituent (C_2). The given attribute is looked for on C_1 or C_2, and its values are copied onto the created constituent as values of the given attribute, except for such values as are part of the definition of the variable, which are not copied. It is possible to generate an attribute with no values using this replacement specification, since all the values from the C_1 or C_2 tags might be among those of the given variable. This specification is especially useful in "erasing" a matched value from a tag specifying a list of

ossible matches. For example, if the C_1 contains GOV/A, B, C and C_2 contains TYPE/B, then the subrule
C_1 . . . GOV/X + C_2 . . . TYPE/X = C_3 . . . GOV/1-X
will create a constituent whose new GOV has the values A, C.

5. The tag replacement has the form ETC/1 or ETC/2 or ETC/1, 2: All the tags from data constituents C_1, C_2, or C_1 and C_2 are copied onto the newly generated constituent with the exception of those whose attributes have been previously mentioned anywhere in the subrule. If the user wishes to copy a tag whose attribute appears previously in the subrule, he must repeat it explicitly in the right half.

1.2.4 Examples of Subrules and Their Application

Let us assume that the left data constituent to be matched against a grammar subrule is:
A SUBCLS/B CASGOV/D ANGOV/NO FORM/N SPECA/SEN
SPECB/REM TYPE/NIS SPECC/LIM SPECD/DEV WORD/F3545

and the right data constituent is:
B SUBCLS/K CASE/D AN/NO FORM/W SPECA/REM SPECB/FIB
TYPE/NIS SPECC/KIM SPECD/GOOB WORD/E2961

Assuming the subrule is
A FORM/X + B SUBCLS/M = C FORM/X SUBCLS/M ETC/2

the subrule will fail, because the tag condition SUBCLS/M is not met, since B is coded with SUBCLS/K. If B were coded with SUBCLS/M, then the subrule would succeed with variable X set to N. The new constituent would be
C FORM/N SUBCLS/M CASE/D AN/NO SPECA/REM SPECB/FIB
TYPE/NIS SPECC/KIM SPECD/GOOB WORD/E2961

Notice that it was necessary to specify SUBCLS/M on the right half as well in order to have the SUBCLS copied. The ETC specification would not copy it, since SUBCLS had already appeared in the subrule.

If the subrule is:
A SUBCLS/B CASGOV/X FORM/Y + B CASE/X FORM/Y =
C SUBCLS/P FORM/Y CASGOV/1-X

then the subrule fails, since Y is set to N from the C_1 constituent, and fails to "agree" on the C_2 constituent. If the C_2 had FORM/N instead of FORM/W then the subrule would succeed with X set to D and Y set to N. The created constituent would be C SUBCLS/P FORM/N CASGOV/* (the * will not print).

If the subrule is:
A SUBCLS/X1-B SPECB/X SPECC/Y SPECD/Z + B ILLEG/*
SPECA/X = C SPECA/Y SPECB/Z SPECC/W1 SPECD/W2

then the subrule fails, since X1 will be first set to B, and then the B will be eliminated from the definition of X1, leaving a null definition, which fails the subrule. The meaning of such a tag condition is "any X1 except B". If A had SUBCLS/C, then the subrule would succeed, with X1 set to C, X to REM, Y to LIM, Z to DIV, and the generated constituent would be:

C SPECA/LIM SPECB/DIV SPECC/W1 SPECD/W2

If the B constituent had the tag ILLEG/PLUS, then the subrule would fail, since the tag condition ILLEG/* requires the absence of tag ILLEG or a null value on it.

1.2.5 Order of Execution of Subrules

For a given pair of parts of speech, control is transferred to a rule packet when two adjacent constituents are found during analysis. Control begins in a rule packet by executing the first subrule in the packet. At the end of execution of a subrule, the subrule has either succeeded (and generated a new constituent), or failed. Flow of control after this point is determined by whether the subrule has succeeded or failed, and what the user has indicated (if anything) in the transfer section of his subrule. The user may specify the next subrule in the packet to be executed in case the subrule succeeds, and the next subrule in the packet to be executed in case the subrule fails, by including a transfer section in his subrule. The transfer section has the form: (S/⟨symbol⟩ F/⟨symbol⟩). The S/⟨symbol⟩ is called the success transfer; the F/⟨symbol⟩ is called the failure transfer. The success transfer and failure transfer may appear in either order; one or the other or both may be absent; if both are absent the enclosing parentheses must also be absent. The ⟨symbol⟩ must appear in the label section of another subrule in the same packet, otherwise the symbol is undefined, and an error message is printed during grammar compilation. Exception: If the symbol is "QUIT", it does not refer to another labelled subrule in the rule packet, but indicates that no more subrules are to be executed. After a subrule is executed, the next subrule to be executed is the one whose label appears in the success transfer of the current subrule if the subrule has succeeded and is the one whose label appears in the failure transfer if the subrule has failed. If the transfer symbol is "QUIT", no more subrules are tried, and the matching process for this pair of constituents is complete. If the transfer symbol has not been specified by the user, control either goes to the next consecutively written subrule in the rule packet, or, if this is the last subrule in the rule packet, the rule is terminated (like "QUIT"). Hence, the user needs to specify labels and transfer symbols only in those cases in which he wishes to depart from the normal sequential execution of the subrules of a rule packet. The user must take care that a subrule does not transfer back to itself and that flow of control does not result in any subrule being executed more than once.

Illustration of flow of control

```
A TY/A ST/X + B TY/K ST/X = C SPEC/FORM ST/X (S/RULE3)
A + B = D
(RULE3) A TY/X + B TY/X = E (F/QUIT)
A + B = D ST/K
A + B = C CL/W
```

The very specific first subrule tests for an exceptional set of tags on A and B and generates a new constituent C. If A and B are not this exceptional case, then control goes to the next consecutive subrule, which has no tag conditions and hence must succeed, producing a D. If the exceptional case has occurred, we do not wish to produce a D, so the S/RULE3 causes a transfer of control to the third subrule, RULE3. Another test is made (for agreement of TY) on A and B. If the subrule succeeds, we wish to generate a new constituent E, but we also wish to generate D ST/K and C CL/W as well. Hence we allow control to continue to the next two subrules with no tag conditions, just in case RULE3 succeeds. If RULE3 fails, however, we do not wish to generate these extra constituents, and, since there are no more subrules in this rule packet, the failure transfer is QUIT.

1.3 CSA System Organization

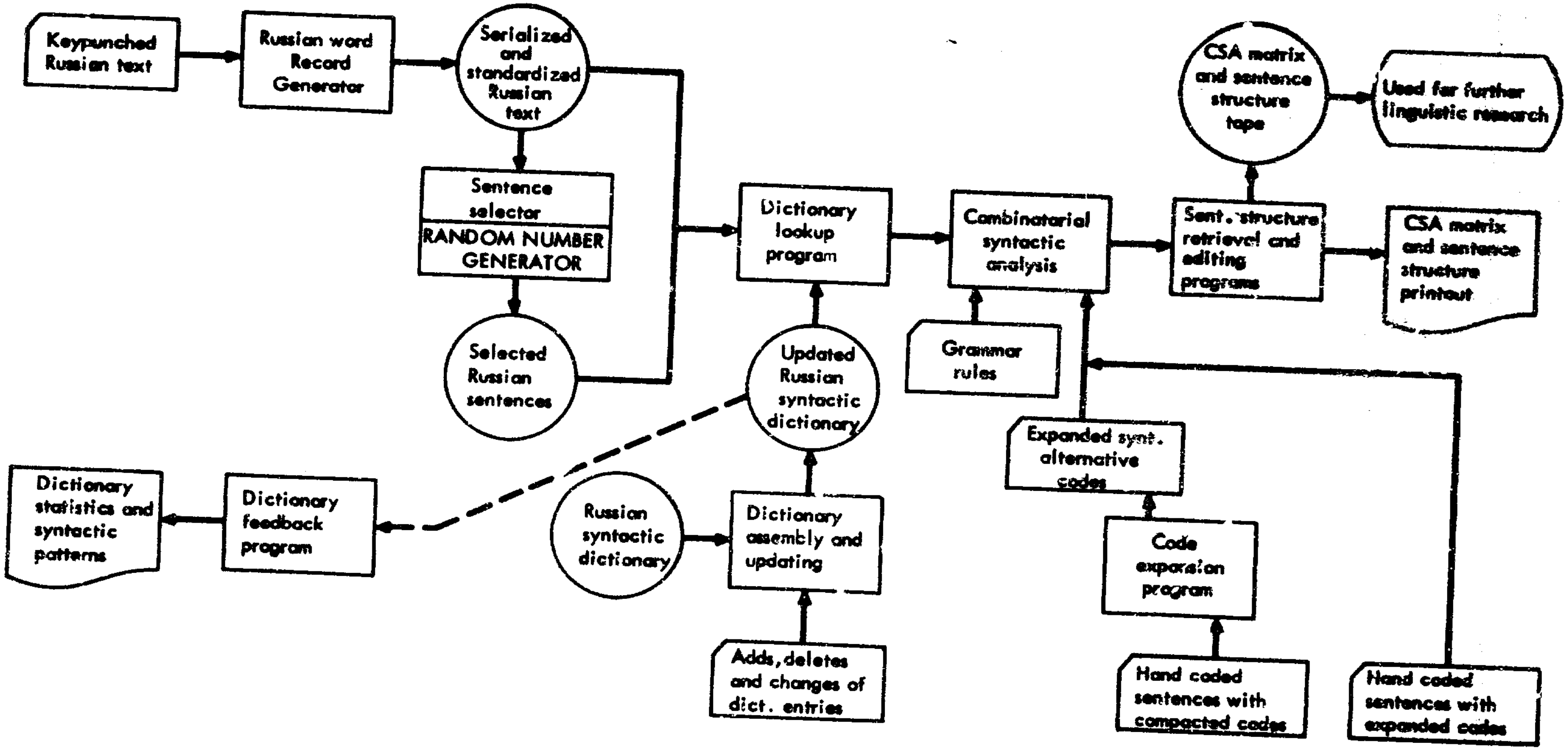
The overall organization of the CSA system is displayed schematically in Figure I-1. The system comprises a parsing package (The Combinatorial Syntactic Analyzer proper), a dictionary assembly/update package, a dictionary lookup routine, and miscellaneous routines for input and output processing and linguistic support. Brief general descriptions of these four components are given below in 1.3.1-1.3.4, respectively. A detailed description of the internal organization of the CSA parsing package is presented in Appendix I-A; the Dictionary Assembly/Update package is described in similar fashion in Appendix I-B.

1.3.1 Combinatorial Syntactic Analysis Routines

The automatic parsing package developed under the present contract consists of twenty-four basic routines in the form of relatively small modules. Each basic operation within the package is assigned to a specific module, so that all direct accesses to a given data storage area used in parsing are made exclusively by the module which has the competence to access that area. This organisation makes it possible to confine the effects of system changes to individual modules, rather than involving the entire program package.

The parsing process is divided into five major steps, the last three of which are repeated for each sentence analyzed:

Figure I-1: COMBINATORIAL SYNTACTIC ANALYSIS: TOTAL SYSTEM CONCEPT AND PROGRAM INTERFACES


Keypunched Russian text
Russian word Record Generator
Serialized and standardized Russian text
Sentence selector
RANDOM NUMBER GENERATOR
Selected Russian sentences
Dictionary lookup program
Combinatorial syntactic analysis
Sent. structure retrieval and editing programs
CSA matrix and sentence structure tape
Used for further linguistic research
CSA matrix and sentence structure printout
Grammar rules
Updated Russian syntactic dictionary
Expanded synt. alternative codes
Code expansion program
Hand coded sentences with compacted codes
Hand coded sentences with expanded codes
Dictionary statistics and syntactic patterns
Dictionary feedback program
Russian syntactic dictionary
Dictionary assembly and updating
Adds, deletes and changes of dict. entries

1. All analysis routines are loaded into memory and the various work areas are laid out.
2. The grammar rules are read in and stored in various tables for subsequent matching against syntactic alternatives of input text words and higher-order constituents.
3. The syntactic alternatives for each word in an input sentence are read into the computer.
4. The sentence is parsed exhaustively according to the CSA algorithm (Section 1. 1), with intermediate results accumulated in a storage matrix.
5. The matrix is searched for complete analyses of the sentence. Each such analysis is printed out in the form of a tree whose nodes consist of constituent names and their associated grammatical tags.

The flow of control of the parsing program can be observed with reference to Figure I-2. After the routines have been loaded in memory and the work areas and tables have been laid out, the main program, which performs the actual parsing, immediately branches to the monitor. The monitor examines all control cards specifying the different tasks of the system, such as compilation of grammar rules, printing out of messages, and reading in of sentences to be analyzed.

Assuming that the first card is a $GRAMMAR control card, the monitor directs all grammar cards to the grammar compiler, whose task is to set up the grammar rules for subsequent matching with the input text constituents. For each card beginning with "*" (a group heading card), the compiler stores the BCD names for the rule constituent pair in unique locations by calling a hash addressing routing HASH, which produces an address pair. This address pair is next stored in an entry in the table of grammar rules RULTAB, the entry address being found by hashing the address pair itself. The RULTAB entry points to the first location in the subrule table TAGTAB where the corresponding subrules will be stored. The rule compiler GRAMAR then compiles all subsequent grammar cards with that heading (the subrule cards) into consecutive locations of TAGTAB in the form of a list structure.

When GRAMAR reaches a control card, it returns to the monitor. If the card contains the command $SENTENCES, control is transferred to the sentence read routine SENTNC, which reads a sentence, each word of which is coded on a word card (signalled by * in column one) followed by cards containing its syntactic alternatives. SENTNC uses HASH to convert the part-of-speech code and tags of each syntactic alternative to the addresses of the locations where their BCD representations are stored and records the addresses in a table in a common storage area.

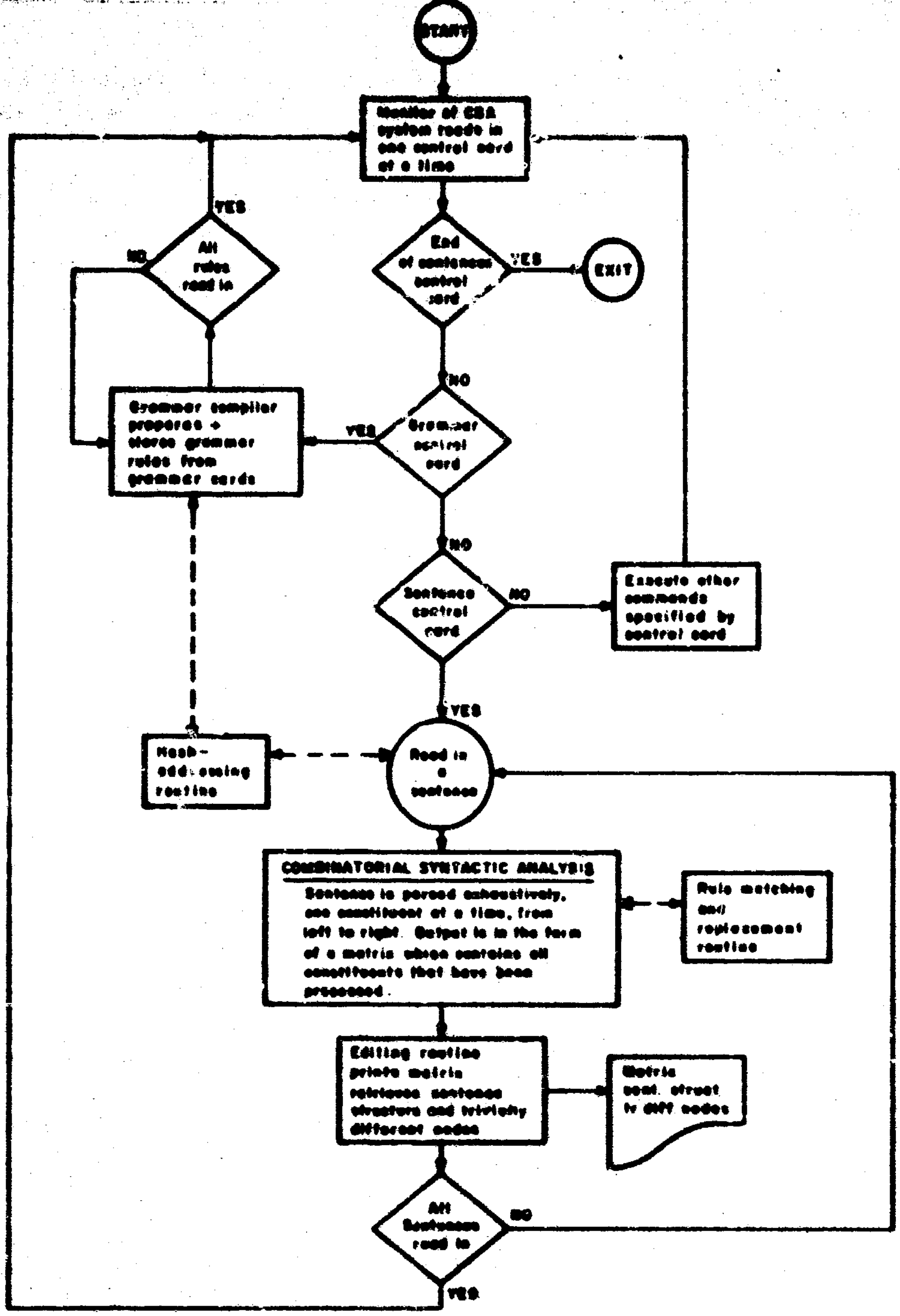

START
Monitor of CSA system reads in one control card at a time
End of sentences control card
YES
EXIT
NO
All rules read in
YES
NO
Grammar compiler prepares + stores grammar rules from grammar cards
YES
Grammar control card
NO
Sentence control card
NO
Execute other commands specified by control card
YES
Hash-addressing routine
Read in a sentence
COMBINATORIAL SYNTACTIC ANALYSIS
Sentence is parsed exhaustively, one constituent at a time, from left to right. Output is in the form of a matrix which contains all constituents that have been processed.
Rule matching and replacement routine
Editing routine prints matrix relations sentence structure and trivially different nodes
Matrix sent. struct iv diff nodes
All Sentences read in
NO
YES

Figure I-2: FLOW OF CONTROL OF CSA

When SENTNC encounters a card with two asterisks separated by a space, it interprets this as an end-of-sentence signal. Control returns to the monitor and from there to the main program ANALYZ, which now is ready to perform the parsing.

Available to this main program is the table in the common storage area whose entries point to the syntactic alternatives. These entries are copied one at a time into a large storage matrix BIGTAB; the program matches all adjacent pairs of constituents against rules in the grammar table, creating a new constituent whenever paired constituent names match and the tag conditions succeed.

The matching procedure is performed by feeding adjacent constituents to a rule matching and replacement routine GETRL which returns (a) the number of higher-order constituents (possibly zero) that can be formed for this pair according to the rules of the grammar and (b) the address of the first of these constituents. The main program ANALYZ copies all these newly created constituents into BIGTAB and proceeds with analysis until the end of the sentence has been reached.

When the main program has finished analyzing a sentence, it transfers to a sentence structure retrieval and editing program PRANS whose task is to retrieve the results of the analysis according to an option control card and print them out. The first operation normally performed by PRANS is to print out the results of the analysis in matrix format by calling a subroutine DBIG which edits and prints each row of the matrix specified in BIGTAB, giving for each constituent its word boundaries, subconstituent addresses, and tags. Next, PRANS scans BIGTAB for constituents which span the entire sentence, and which have constituent names agreeing with a set of permissible sentence structure nodes. PRANS then prints out the tree for the sentence nodes which it locates, recording the constituent name and tags for each node. According to an option which may be selected by the user, trees representing the distinctive portions of analyses suppressed because of their partial duplication of previous analyses may also be retrieved and printed out.

After the PRANS has processed all the permissible analyses, a test is made to determine whether more sentences are to be processed. If so, control returns to the sentence read program, and the process continues as before. If no sentences remain, control is transferred to the monitor. A final exit is made when the monitor reads a $EXIT control card.

1.3.2 Dictionary Assembly/Update Package

The Dictionary Assembly/Update package consists of eight basic modules as well as the Sort/Merge system and input/output tape control routines. The package was designed in modular fashion in order to facilitate modification of and additions to the total system. As in the CSA system, each basic operation within the program is assigned to a specific module.

The system is entered through the transfer vector MONITOR. The CONTROL routine handles the flow of control within the total assembly framework and READP interprets and prints the user's parameters, specifying the type of updating, type of printout, etc. CARDRD reads the input cards obtained from the lexicographer, which consist of entries for new words with their syntactic alternatives, entries to be deleted, and entries to be changed. ASSLY, an 'assembly' routine in the strict sense of the word, collects these input parts into pre-dictionary records and converts Cyrillic characters into machine-coded sortable bytes. The task of the Sort/Merge system is to sort these records alphabetically on the words and direct them to the updating routine UPDATE, which -- according to the user's parameters -- adds deletes, and/or changes the dictionary entries, thereby creating an updated version of the current dictionary. Also, according to options, the updated dictionary and the added, deleted, and changed records may be edited and printed out by DICPNT for human investigation. The eighth module INTER, serves as a common boundary between the Sort/Merge system, the input/output tape control routines, and the remaining seven modules.

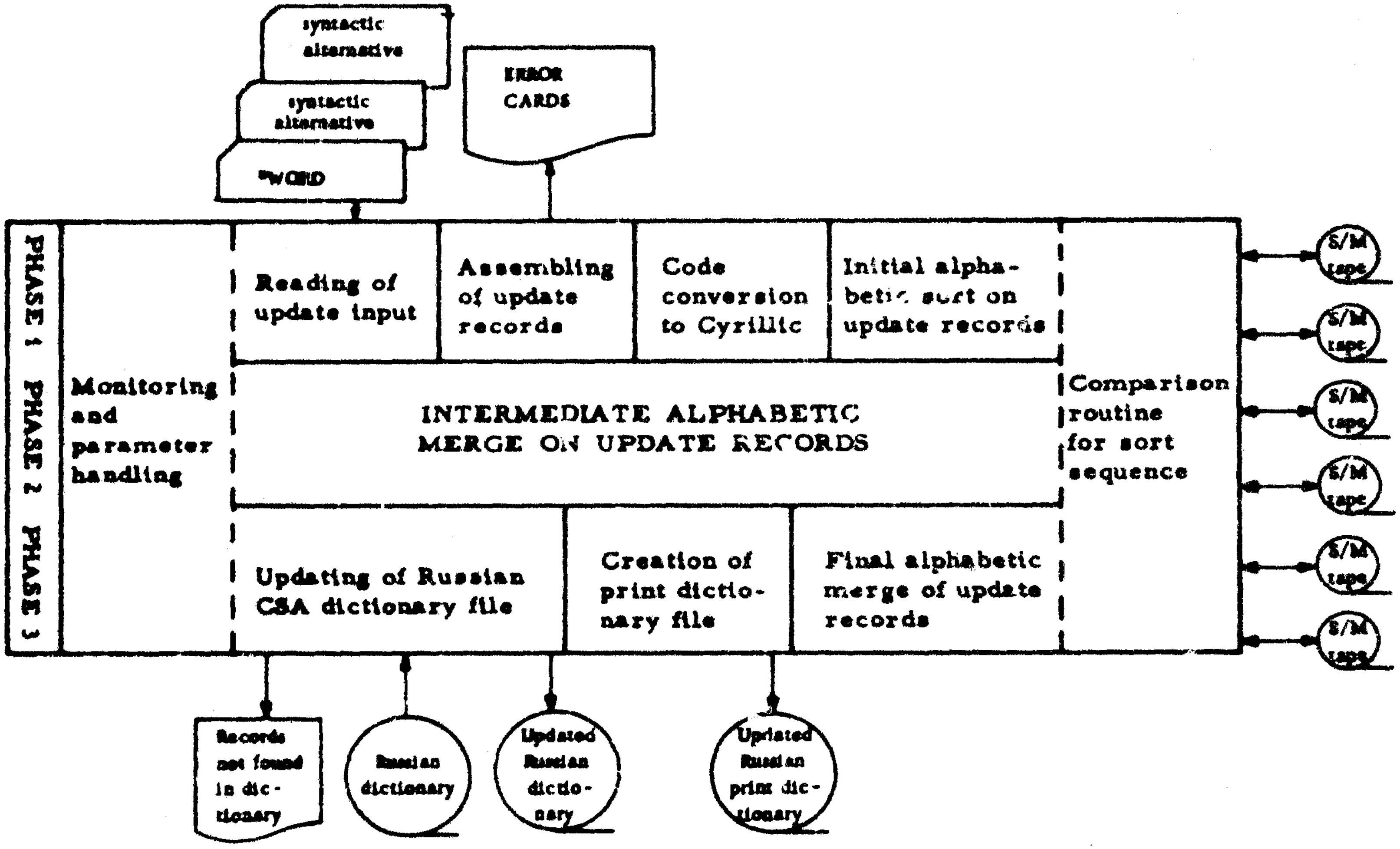
A dictionary entry consists of several logical machine records, the first of which contains the word and some bookkeeping information, such as the date of acquisition, name of lexicographer, etc. The succeeding logical records contain the syntactic alternatives of the word.

For the Assembly/Update package a Sort/Merge system (Grove, 1967) is utilized which avoids the need to write the input data set (update records) on a tape before it can be sorted. In addition, substantial parts of the sorting operation are overlapped with operations of the Assembly/Update package in which the Sort/Merge system is embedded (Figure I-3).

The update records to be sorted are assembled and transferred by ASSLY to the initial sort phase without every being placed on a physically distinct update-record file. This initial phase of the Sort/Merge system is actually time-shared with ASSLY. The final merge phase also does not create a physically distinct update-record file. Instead, the alphabetically sorted records are called, one by one, by the UPDATE module. The latter matches them against the old dictionary and modifies it accordingly, thereby creating an updated version of the dictionary file. It is important to note that the desired sequence of the sort is specified by a comparison routine SORT belonging to the Assembly/Update package rather than to the Sort/Merge system. This comparison routine is called by the Sort/Merge system whenever two update records are to be compared.

Figure I-3. FLOW OF CONTROL OF DICTIONARY ASSEMBLY/UPDATE PROGRAM

(Operations between horizontal lines are performed concurrently. Intermediate merge on update records may be skipped if dataset is small. The sort comparison and monitoring routines are used in all three program phases.)


syntactic alternative
syntactic alternative
WORD
ERROR CARDS
PHASE 1
PHASE 2
PHASE 3
Monitoring and parameter handling
Reading of update input
Assembling of update records
Code conversion to Cyrillic
Initial alphabetic sort on update records
INTERMEDIATE ALPHABETIC MERGE ON UPDATE RECORDS
Comparison routine for sort sequence
Updating of Russian CSA dictionary file
Creation of print dictionary file
Final alphabetic merge of update records
S/M tape
S/M tape
S/M tape
S/M tape
S/M tape
S/M tape
Records not found in dictionary
Russian dictionary
Updated Russian dictionary
Updated Russian print dictionary

1.3.3 Dictionary Lookup Program

This program performs dictionary lookup by comparing Russian text words with a Russian dictionary. The program reads input sentences from tape and creates logical records of one text word each. These records are alphabetically sorted and serially looked up in a tape dictionary. The output of the dictionary lookup is then resorted to conform with the word order of the original input sentences. The resorted looked-up text is now ready to serve as input to the parsing process.

The Sort/Merge system (Grove, 1967) employed by the Assembly/Update package is also used by the dictionary lookup program. The result is that substantial parts of the Sort/Merge operations are overlapped with matching operations of the dictionary lookup program. Since the Russian word records must be sorted alphabetically, matched against the dictionary, and then resorted into their original text order, the last phase of the first sort, the dictionary lookup proper, and the first phase of resorting into text order all occur concurrently (Figure I-4).

1.3.4 Miscellaneous

Random number generator

This program is designed to produce a series of unique random numbers according to specified limitations.

A collection of n random numbers is generated where n is defined by the user's parameter AMOUNT. These random numbers may range from 1 to a maximum number $m \leq 32000$ which must also be defined by a user's parameter RANGE. If the quantity of random numbers and/or the range is to be changed, the corresponding parameters must be modified accordingly.

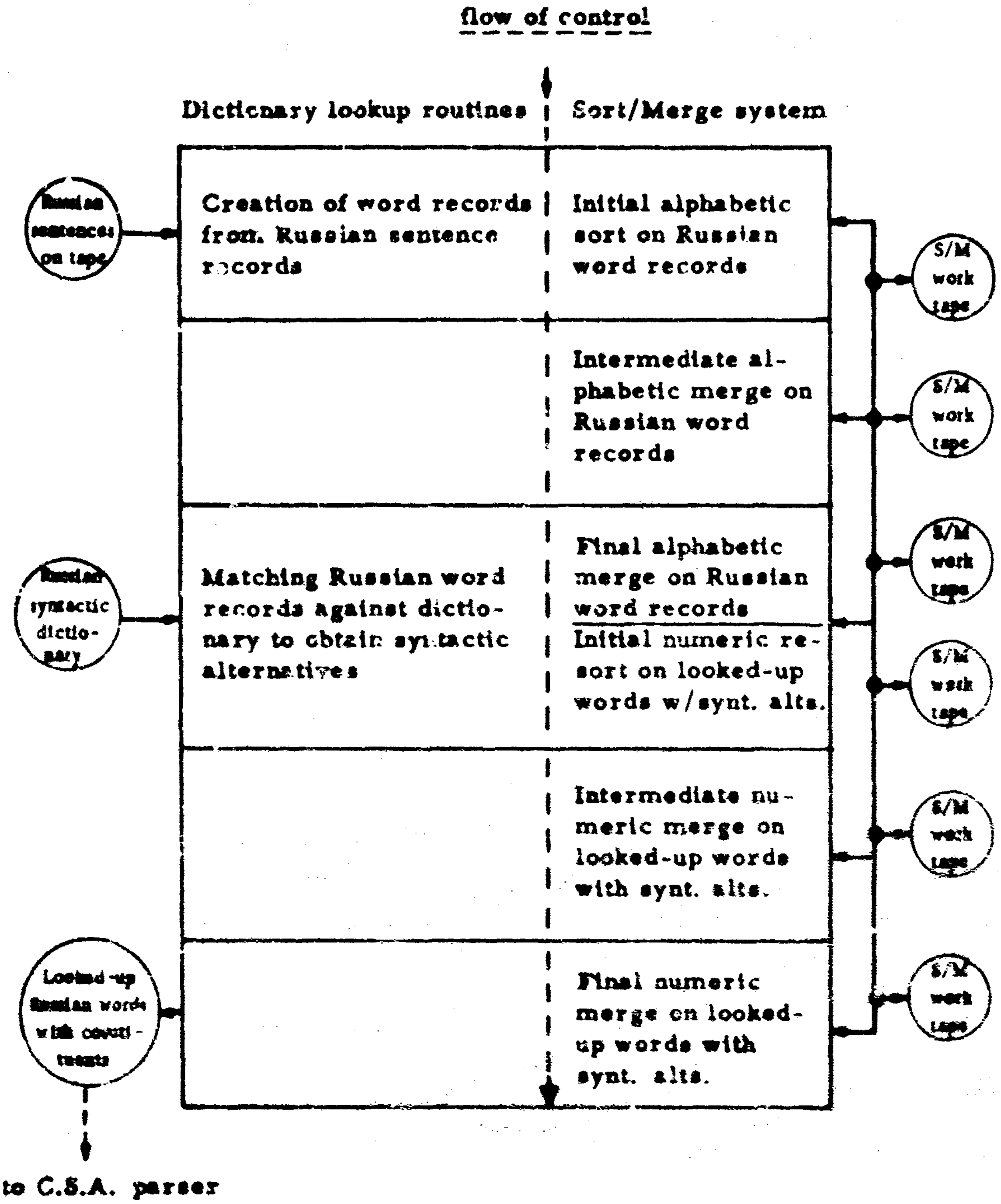
After the generation of each random number, the number is checked against previous results to determine whether or not it is identical to a number that has already been created. If it is, a new random number is generated, thereby assuring uniqueness of the numbers. In our applications, the random number generator is inserted as a subroutine in a Russian sentence selector program which serves to extract a sample of n Russian sentences from a population of m sentences. The sample sentences can then be processed by the Combinatorial Syntactic Analysis system.

Code expansion program

In order to facilitate coding of text words which are associated with long strings of tags, a compacted code may be assigned to the word instead of the detailed part-of-speech and tag string. A program then converts these

Figure I-4: FLOW OF CONTROL OF THE DICTIONARY LOOKUP PROGRAM

(Operations between horizontal lines are performed concurrently.)


flow of control
Dictionary lookup routines
Sort/Merge system
Russian sentences on tape
Creation of word records from Russian sentence records
Initial alphabetic sort on Russian word records
S/M work tape
Intermediate alphabetic merge on Russian word records
S/M work tape
Russian syntactic dictionary
Matching Russian word records against dictionary to obtain syntactic alternatives
Final alphabetic merge on Russian word records
S/M work tape
Initial numeric resort on looked-up words w/synt. alts.
S/M work tape
Intermediate numeric merge on looked-up words with synt. alts.
S/M work tape
Looked-up Russian words with constituents
Final numeric merge on looked-up words with synt. alts.
S/M work tape
to C.S.A. parser

"shorthand" codes into the expanded mnemonic syntactic alternative codes (constituent name and tag string) accepted by the Combinatorial Syntactic Analysis program. This is accomplished by matching the compacted codes against a dictionary which contains these codes as match arguments and expanded mnemonic alternatives as output functions.

Russian word record generator

This program accepts keypunched Russian text as input and generates serialized sentence records which serve as input either to the dictionary lookup program of the CSA system or to that of the predictive Russian Syntactic Analyzer (Section III). (As indicated in Figure I-1, the user has the option of employing the sentence selector to extract a random sample from the set of sentence records prior to the dictionary lookup phase.)

Dictionary feedback program

This program serves to inform the linguist about the status of the CSA dictionary at any given time. By sorting the dictionary file on certain control fields, it can (1) produce a list of all unique syntactic alternatives in the file, thereby displaying all existing combinations of a part-of-speech code with a tag string, and (2) retrieve all alternatives having any combination of part of speech, tags, or tag components specified by the linguist. Whatever options are selected, the program always writes out counts of all master records, all alternative records, and all uniquely patterned alternatives in the dictionary.

REFERENCES

Chomsky, N. (1965) *Aspects of the Theory of Syntax.* Cambridge, Mass.: M.I.T. Press.

Grove, M. C. (1967) *Sort/Merge Subroutine Packages for 7090/4 and 7040/4.* Research Report RC-1779. Yorktown Heights, New York: International Business Machines Corporation.

Kuno, S. (1965) Personal communication.

Plath, W. J. (1963) "Multiple-path Syntactic Analysis of Russian", *Mathematical Linguistics and Automatic Translation,* Report No. NSF-12. Cambridge, Mass.: Computation Laboratory of Harvard University.

Sakai, I. (1962) "Syntax in Universal Translation", *Proceedings of the 1961 International Conference on Machine Translation of Languages and Applied Linguistic Analysis.* London: Her Majesty's Stationery Office.

Yngve, V. H. (1961) *An Introduction to COMIT Programming.* Cambridge, Mass.: The Research Laboratory of Electronics and the Computation Center, M.I.T.

APPENDIX I-A: C.S.A. Program Logic Manual

Table of Contents

The C.S.A. programs are grouped according to their logical functions.

ROUTINES FOR DECK NAMED TAPEIO

System entry points: TAPOPN, TAPGET, TAPPUT

Deck: TAPEIO
Routine: TAPPUT
Type: CSA-independent
I/O type dependent
Entry point
Calling sequence: TSX TAPPUT, 4
PZE WHAT, , HOWMUCH
Function: Puts out a physical record of length HOWMUCH starting from location WHAT onto the output tape.
Restrictions: HOWMUCH ≤ internal buffer size (now 22). Most tape systems require output record to be at least three words long so as not to be recognized as a "noise record".
Operation: Calls CHECK to complete last I/O. Then copies user buffer into internal buffer. Calls EMPTY to start emptying internal buffer.

Deck: TAPEIO
Routine: FILL
Type: CSA-independent
I/O type dependent
Internal to TAPEIO
Calling sequence: TSX FILL, 4
Function: Starts filling the internal read buffer.

Deck: TAPEIO
Routine: EMPTY
Type: CSA-independent
I/O type dependent (tape)
Internal to TAPEIO
Calling sequence: TSX EMPTY, 4
Function: Starts emptying the internal write buffer.

Deck: TAPEIO
Routine: CHECK
Type: CSA-independent
I/O type dependent (tape)
Internal to TAPEIO
Calling sequence: TSX CHECK, 4
Function: Makes sure that last I/O has finished. It then checks for end-of-file, redundancy, end-of-tape conditions. If redundancy has occurred, retries up to 20 times.

If end-of-tape, rewinds and unloads tape. Sets switches in either read or write buffer (whichever was used last) to indicate either normal completion or one of the above unusual ends. Then it exits.

Deck: TAPEIO
Routine: TAPOPN
Type: CSA-independent
I/O type dependent (tape)
Entry point

Calling sequence:

```
        CAL   BUFWD
        LDQ   DEVAD
        TSX   TAPOPN, 4

BUFWD   PZE   BUFFER, , LENGTH
DEVAD   PZE   , , DEV
```

(AC sign + to open input, - to open output)

Function: Must be called to initialize TAPGET and TAPPUT. Defines an internal input (or output) buffer specified by BUFWD, and specifies DEV as the address of the tape device used for input (or output). (DEV is the octal tape address). For input, the BUFFER must be three words longer than actual input buffer size desired (to accommodate switches). For output, BUFFER must be one word longer than desired. Hence, for an input buffer of 14, and output of 22, buffer sizes of 17 and 23 must be given, respectively.

Deck: TAPEIO
Routine: TAPGET
Type: CSA-independent
I/O type dependent (tape)
Entry point

Calling sequence:

```
TSX   TAPGET, 4
PZE   USERBF, , LENGTH
. . . end-of-file return. . .
. . . redundancy return. . .
. . . normal return. . .
```

Function: Reads one physical record into user's buffer. (Used for line-input.)

Restrictions: Specified length ≤ internal buffer length (now 14). Calls CHECK to complete last I/O, if any. Checks switches to see if last read was end-of-file or redundant -- if so, exits appropriately. (If there was only 1 redundancy, it retries until either a normal return can be made or it gets 20 redundancies in a row, in which case a redundancy exit is made.) On completed operation, copies internal buffer with next record.

ROUTINES FOR DECK NAMED FILE

System entry points: PUT CARDRD SET SETS BCDCAN DEVCAN BUFFER

Deck: FILE
Routine: PUT
Type: Configuration-independent
Entry point
Calling sequence:
```
TSX    PUT,4
PZE    WHAT,,HOWMUCH
```
Function: Puts out a record of length HOWMUCH starting from location WHAT onto the output device. Which device it is depends upon previous calls into deck FILE defining the output device.
Operation: Routine SETS has set PUT as a transfer to the appropriate already-initialized device "PUT" routine (e.g., TAPPUT, DSKPUT, etc.). Control goes directly to that routine.

Deck: FILE
Routine: CARDRD
Type: Configuration-independent
Entry point
Calling sequence:
```
TSX    CARDRD,4
PZE    USERBF,,LENGTH
...end-of-file return...
...redundancy return...
...normal return...
```
Function: Reads one card image record into user's buffer from input device.
Operation: SETS has set CARDRD to be a transfer to the appropriate device GET routine, which has the same calling sequence (e.g., TAPGET).

Deck: FILE
Command: SET
Type: Configuration-independent
File system routine
Command entry point

Command card format: $ SET file-name TO device-name

Function: File-name is a name of the user's choosing. Device-name must be one specified by device table (deck: DEVICE). The device associated with the device-name is now associated with the user's file-name, in that calling BCDCAN with this file-name in the future will obtain the "canonical" device address. File names INPUT and OUTPUT are treated specially, for they refer to the system INPUT and OUTPUT files, respectively. When these file-names are set, the device descriptions of the corresponding device units are obtained, the routine CARDRD or PUT is set to point to the appropriate device routine, and the device's "open" entry is called. Hence, $ SET INPUT TO A6 would set CARDRD to the TAPGET routine, and would call TAPOPN with the appropriate parameters from the device description of A6.

Deck: FILE
Routine: SETS
Type: Configuration-independent
Entry point
Calling sequence:

```
CAL    file-name
LDQ    device-name
TSX    SETS, 4
```

Function: Simulates the execution of the command $ SET file-name TO device-name.

Deck: FILE
Routine: BCDCAN
Type: Configuration-independent
Entry point
Calling sequence:

```
CAL    file-name
TSX    BCDCAN, 4
```

Function: Returns in the AC the 15-bit "canonical address" of the device possessing the given file-name. A file-name is given a device by means of a SET card or call to SETS.

Operation: Simply looks up the file-name in a table called CANTAB in the deck DEVICE, and returns the address of the corresponding device.

Deck: FILE
Routine: DEVCAN
Type: Configuration-independent
Entry point

Calling sequence:

```
CAL    device-name
TSX    DEVCAN, 4
```

Function: Returns in the AC the 15-bit "canonical address" for the device whose printname in BCD appears in the AC.

Operation: Simply looks it up in DEVTAB, which the user must have assembled with the system.

Deck: FILE
Routine: BUFFER
Type: Configuration-independent
Entry point

Calling sequence:

```
CAL    LENGTH, , NAME
TSX    BUFFER, 4
```

Function: Returns in the AC the buffer control word associated with the 15-bit NAME specified in the calling sequence. A buffer control word is of the form: PZE origin, , length. If there has not yet been a buffer associated with this name, a new buffer is fetched from available space (getting available core space is a system configuration dependent process), and its control word is returned in the accumulator.

Operation: Scans a table, BUFTAB, which associates NAMEs with buffer origins (the lengths are the same as the LENGTH specified by calling sequence). If a buffer with the requested NAME is found, its address and user's LENGTH are returned. If not, the next free space in available buffer storage is obtained, an entry is created in BUFTAB, and the appropriate buffer control word is returned.

Restrictions: At present, buffers are obtained from a pool of maximum size 120 -- enough for 3 input and 3 output line buffers. FILE system uses addresses of device-table entries as "names" of buffers.

DECK NAMED DEVICE

System entry points: DEVTAB CANTAB

Deck: DEVICE
Table: DEVTAB
Format: 2-word entries terminated by fence of PZE

```
        BCI          1, device-name
        PZE          "canonical address"
```

The "canonical address" is a pointer to a device description. For input line files, the "device description" is a two-word entry:

```
PZE    open routine,, buffer length needed
PZE    get/put routine,, physical address
```

Deck: DEVICE
Table: CANTAB
Format: CANTAB is set by calls to SETS (other than with file-name INPUT or OUTPUT), and tested by BCDCAN. Entries are stored by SETS in the form:

```
BCI          1, file-name
PZE          "canonical address"
```

terminated by fence of zeroes.

ROUTINE FOR DECK NAMED <u>SNTAPE</u>

System entry points: SNTAP

Deck: SNTAPE
Routine: SNTAP
Type: 7094 tape I/O dependent
Entry point
Calling sequence:

```
TSX    SNTAP, 4
PZE    USERBF,,RETURN
...end-of-file return...
...redundancy return...
...normal return...
```

Function: Reads one logical record from tape A5 into user's buffer. Tape A5 contains blocked records of maximum physical length 500, each of whose logical records is prefixed by a logical control word containing PZE relative address of the control word of the next logical record,, number words of current logical record.
Restrictions: USERBF must be large enough to accommodate the largest-sized logical record which has been put onto tape A5.

ROUTINES FOR DECK NAMED <u>PRINT</u>

System entry points: CHAROU FLUSH SETPUT EDIT EDITX
INTPUT VARPUT ALFPUT BLNPUT SETOVF

Deck: PRINT
Routine: CHAROU

Type: Configuration-independent
Entry point

Calling sequence:

```
TSX   CHAROU, 4
PZE   FROM,, BITCT
```

Function: Moves BITCT number of bits from FROM address into print line buffer (which must have been provided by SETPUT). If this causes an overflow past the "bell" on the line, only those bits up to the bell position are inserted, the overflow routine (set by SETOVF, or a standard overflow routine by default) is called, and the remaining bits are then inserted (the overflow routine has presumably made this possible). The current setting of the bell is after the 120th character position. CHAROU does not know this. The bell position is defined with the print buffer by SETPUT.

Deck: PRINT
Routine: FLUSH
Type: Configuration-independent
Entry point

Calling sequence:

```
TSX   FLUSH, 4
```

Function: The print line which had been set up by CHAROU is now put out on the output device by calling PUT. Before this can be done, the last word must be padded on the right with blanks, and if there are fewer than three words, blank words must be inserted, so that the record put out by PUT is three words or longer (lest it look like a "noise record"). Then the CHAROU routine is reinitialized to start sorting characters into bit position 1 again.

Deck: PRINT
Routine: SETPUT
Type: Configuration-independent
Entry point

Calling sequence:
(at present)

```
        CAL   POINT
        TSX   SETPUT, 4
POINT   PZE   BUFBL
```

Function: Initializes the routine CHAROU by providing a buffer. To initialize a buffer, of length 22, with "bell" after 20, BUFBL should be:

```
BSS   5                  (words used by CHAROU)
PZE   address overflow routine
PZE   Line+20,, 20
PZE   Line+22,, 2
PZE    20,,36         (decrement always 36)
PZE    Line+20,,20    (always identical to word 6)
```

Deck: PRINT

Routines: EDIT, EDITX

Type: Configuration-independent
Entry point

Calling sequence:

```
TSX    EDIT or EDITX, 4
```

(see R.C.C. manual for commands)

Function: Processes the commands defined for EDIT as defined in the IBM Research Center Computing Manual --the restricted class of permitted commands includes BC., BK., IN., and OC.. The differences between the function of this routine and the function of the IBM EDIT are: Output is sent, via PUT, to whatever device the I/O system is using for output. The I/O system also checks for end of output tape. Integer specification prints unsigned numbers -- no space is needed for a sign. The OC specification is not used for octal -- it is used for a new kind of integer specification whereby the length need not be defined by the user -- the number of significant digits is used as the length.

Operation: All EDIT commands end with either W (write print line on output device), E (terminate calling sequence), N (neither), or B (both). In addition, the command specifies the appropriate format conversion. The EDIT subroutine in this system takes each individual edit command, determines the effective address (from actual address and index register if specified), and then calls either BLNFUT, ALFPUT, VARPUT, or INTPUT, depending on the command. If the command was for write print line. FLUSH is called after executing the command. If the command specified E for end calling sequence, the routine returns to the caller.

Deck: PRINT

Routine: INTPUT

Type: Configuration-independent
Entry point

Calling sequence:

```
TSX    INTPUT, 4
PZE    NUMBER,,LENGTH
```

where NUMBER is the address of an integer less than 1,000,000 and LENGTH is the number of print positions to be used.

Function: Inserts the specified integer into the print line, using LENGTH number of print positions. The integer is padded on the left with blanks, if necessary.

Operation: Calls BINBCD to convert integer. Then pads with blanks. Then calls CHAROU to put out converted integer, making sure that the bit count is 6 times the length count of characters.

Deck: PRINT

Routine: VARPUT

Type: Configuration-independent
Entry point

Calling sequence:

```
TSX    VARPUT, 4
PZE    NUMBER
```

Function: NUMBER is the address of an integer less than 1,000,000. The significant digits of this number are put into the print line. No additional blanks are inserted.

Operation: BINBCD is called to convert the integer. The XR2 will tell how many significant digits there were, and CHAROU is then called to put out 6 times that number of bits.

Deck: PRINT

Routine: ALFPUT

Type: Configuration-independent
Entry point

Calling sequence:

```
TSX    ALFPUT, 4
PZE    WHERE, , COUNT
```

Function: A COUNT number of characters is entered into the print line from the location WHERE (and beyond, if more than 6 characters are entered).

Operation: CHAROU is simply called with a bit count of 6 times COUNT.

Deck: PRINT

Routine: BLNPUT

Type: Configuration-independent
Entry point

Calling sequence:

```
TSX    BLNPUT, 4
PZE    , , NUMBER
```

Function: Puts NUMBER blanks into a print line.

Operation: Puts 6 times NUMBER bits of blanks into print line with CHAROU.

Deck: PRINT
Routine: SETOVF
Type: Configuration-independent
Entry point
Calling sequence: TSX SETOVF, 4
PZE RTN (zero for resume "default" overflow routine)

Function: The next time a call to CHAROU causes bits to be entered past the "bell" on a line, the routine RTN is TSXed to. If no overflow routine is specified, the program will TSX STAND, 4 and the routine STAND will simply call FLUSH to terminate the current line, and call CHAROU with a single blank to insert into the carriage control position of the next line.

ROUTINE FOR DECK NAMED <u>BINBCD</u>

System entry point: BINBCD

Deck: BINBCD
Routine: BINBCD
Type: Configuration-independent
Entry point
Calling sequence: CLA NUMBER
TSX BINBCD

Function. The AC contains an unsigned number less than 1,000,000. The result is 6 BCD digits in the AC in printable form. Index register 2 will contain the number of significant digits of the result. Non-significant zeroes are not blanked out -- the caller may blank them out if he wishes, since index register 2 already contains the number of significant digits.

ROUTINES FOR DECK NAMED <u>MONIT</u>

System entry points: RETURN READIN STACK COMMAN RNEXT
CALL DATSAV DSTACK

General description: READIN reads a card (using SENTRD) and goes to routine specified by first word after $ of that card. $ must be in column 1, or the card will be skipped. STACK holds the arguments on the card. COMMAN

Deck: PRINT
Routine: SETOVF
Type: Configuration-independent
Entry point
Calling sequence:
```
TSX    SETOVF, 4
PZE    RTN (zero for resume "default"
            overflow routine)
```
Function: The next time a call to CHAROU causes bits to be entered past the "bell" on a line, the routine RTN is TSXed to. If no overflow routine is specified, the program will TSX STAND, 4 and the routine STAND will simply call FLUSH to terminate the current line, and call CHAROU with a single blank to insert into the carriage control position of the next line.

ROUTINE FOR DECK NAMED BINBCD

System entry point: BINBCD

Deck: BINBCD
Routine: BINBCD
Type: Configuration-independent
Entry point
Calling sequence:
```
CLA    NUMBER
TSX    BINBCD
```
Function: The AC contains an unsigned number less than 10,000. The result is 6 BCD digits in the AC in printable form. Index register 2 will contain the number of significant digits of the result. Non-significant zeroes are not blanked out -- the caller may blank them out if he wishes, since index register 2 already contains the number of significant digits.

ROUTINES FOR DECK NAMED MONIT

System entry points: RETURN READIN STACK COMMAN RNEXT CALL DATSAV DSTACK

General description: READIN reads a card (using SENTRD) and goes to routine specified by first word after $ of that card. $ must be in column 1, or the card will be skipped. STACK holds the arguments on the card. COMMAN transfers to routine specified by $ card in STACK assumed to have already been read in. RNEXT reads next card and goes to routine specified by $ card. RETURN goes back to routine which last called READIN. CALL first copies user's stack into STACK, and from there on acts like COMMAN. Needs table COMTAB containing pairs of entries BCI 1, NAME PZE entry point. This is in deck COMMAN. DATSAV moves the date information from the $DATE card, which was temporarily copied into STACK, into DSTACK for any date information printings.

Deck: MONIT

Routine: CALL

Type: Configuration-independent
Monitor
Entry point

Calling sequence:
```
TSX    CALL, 4
PZE    WHENCE
```

Function: Copies the user's stack, starting from WHENCE, into the monitor STACK. Then prints STACK via PRBCD. Then goes to COMMAN to execute the command.

Restriction: The monitor STACK is currently only of length 24.

Deck: MONIT

Routine: READIN

Type: Configuration-independent
Monitor
Entry point
Needed for CSA monitor operations

Calling sequence:
```
TSX    READIN, 4
```

Function: Initial entry to monitor. (OPENIO must have been called first.) Skips to a $ card, then (at COMMAN) transfers control to routine specified by $ card. The card is saved in STACK for reference by the routine. Control is returned to user when some program calls TRA RETURN.

Operation: Calls SENTRD to read into STACK. When word at STACK is $, takes STACK + 1 and scans down COMTAB for matching BCD word, which must be found, or else an error message is given and the job is killed. When the word is found, the corresponding entry point is transferred to. The Index register 4 is saved so that RETURN can give control back to the

caller.

Deck: **MONIT**
Routine: **DATSAV**
Type: **Configuration-independent**
Monitor
Entry point
Calling sequence: **None**
Function: **When a $DATE card is read, the monitor transfers via the COMTAB entry**

```
BCI     1, DATE
PZE     DATSAV
```

to this routine which moves the date information from the temporary STACK into the permanent DSTACK from which it can be printed anytime by the DATE routine.

DECK NAMED COMMAN

System entry point: COMTAB

Deck: COMMAN
Table name: COMTAB
Type: Used by MONIT -- must be present on all jobs with monitor
Configuration-independent
Table
Format: A list of two-word entries, terminated by a fence of zeroes. Each entry is of the form:

```
BCI          1, NAME
PZE          TRANSFER POINT
```

Function: To use the monitor system, supply a set of COMTAB entries, each containing as NAME the first 6 characters of the name of the desired command, and as transfer point the address of the routine to process the appropriate command.

Operation: When the monitor is in control, $ cards are read into STACK (an entry point in MONIT). The word after the dollar sign is looked up in COMTAB and control is then transferred to the entry point corresponding to the matched COMTAB entry. Other parameters may appear on the $ card, and the monitor may interrogate these parameters by referencing the STACK. STACK itself will contain a $, STACK + 1 the name of the command, and the remaining locations will contain the parameters.

ROUTINES FOR DECK NAMED READ

System entry points: SENTRD OPENIO

Deck: READ
Routine: OPENIO
Type: 7094/configuration-dependent
Entry point
Calling sequence: TSX OPENIO, 4
Function: This routine must be called to initialize input-output for the particular system configuration at a given installation. At the IBM Research Computing Center, OPENIO interrogates sense switch 1. If it is on, the routine calls SETS to define INPUT file as A2 device, OUTPUT as A3; if it is off, the routine defines INPUT file as DISKIN and OUTPUT as DISKOU. These devices must appear in the DEVICE deck's table, if the appropriate sense switch setting is used. Since at the moment only A2 and A3 (and other tapes) appear in DEVICE table, sense switch 1 can only be on.
Operation: Checks the sense switch settings, and calls SETS in FILE deck with appropriate device names. Calls SETPUT in PRINT to provide a print line buffer for use by EDIT (and any routines calling CHAROU).

Deck: READ
Routine: SENTRD
Type: Configuration-independent
Entry point
Calling sequence: TSX SENTRD, 4
PZE STACK,, SIZE
Function: The user can specify a STACK of any size he wishes. SENTRD will read a "logical card". A logical card runs from column 1 through column 72, unless column 72 is punched with an 11-punch (minus sign), in which case it includes columns 1-71 plus continuation from the next physical card. Any number of physical cards can form a logical card as long as a "-" in column 72 indicates that the next card is a continuation. SENTRD looks for strings of consecutive alphabetic characters, separated by blanks or by "break characters" (comma, dollar sign, period, parentheses, slash, equal sign). Each string of alphabetic characters between blanks or breaks, as well as each individual break character, is stored, left justified, padded with blanks, in the next free location in STACK.

If a string is longer than 6 characters, the components of the string are separated by a "logical concatenator" symbol consisting of the word 767676767676. Blanks from cards, which are used only to delimit fields of text characters, are not put into the STACK. After the last word has been stored in STACK, a fence of 777777777777 is stored. SIZE is the maximum number of words (including fence) which will be stored in the STACK. Overflow will be lost, and the fence stored in the last location, if the logical card has too many fields.

Example: Suppose the logical card contains:

ABC DEF K/L C, D=EFGHIJK

The stack will contain:

ABC
DEF
K
/
L
C
,
D
=
EFGHIJ
767676767676
K
777777777777

ROUTINES FOR DECK NAMED <u>HASHTG</u>

System entry points: HASHTG PUTTAG CLEAR CANCEL

General description: These routines are used to develop tags created either as a result of reading the codings of the input items or as a result of creation by the GETRL program for creating new constituents. In either case, the tags are generated linearly, starting with the first attribute, followed by its value(s), and so on. Each time one wishes to append an 18-bit tag element to the list of tags currently being generated, one calls PUTTAG. When one has completed an entire data constituent after calling PUTTAG a number of times, one then calls HASHTG and gets the address of this newly created "data constituent". To ignore previous calls to PUTTAG since the last call to

HASHTG, call CANCEL. Thereafter, the next call to PUTTAG will start at the beginning of a new constituent. To clear the TAGS table stored in HASHTG, call CLEAR.

Deck: HASHTG
Routine: PUTTAG
Type: Configuration-independent
Entry point
Calling sequence:

```
CAL    component
TSX    PUTTAG, 4
```

where component is of the form:

```
PZE    18-bit tag component
```

Function: PUTTAG appends this 18-bit tag component to the other 18-bit tag components created since the last call to HASHTG or CANCEL. When HASHTG is called, the string of 18-bit components stored via PUTTAG will be stored away and the address of the origin of the data constituent will be returned by HASHTG.

Operation: PUTTAG uses the table TEMPTG (size 100) to store these 18-bit quantities, half-word by half-word. It keeps a running checksum of these quantities, to be used by HASHTG as a "hash sum".

Deck: HASHTG
Routine: HASHTG
Type: Configuration-independent
Entry point
Calling sequence:

```
TSX    HASHTG, 4
```

Function: If PUTTAG has been called one or more times since the last call to HASHTG or CANCEL, HASHTG inserts the stacked up tags (which have been put 18 bits at a time onto TEMPTG by PUTTAG) into the tag table, TAGS (length 8000), provided that the same string has not already been stored. In either case, HASHTG returns in the AC address field the address of the beginning of this tag string. AC sign is "-" if tag has previously been stored in TAGS.

Operation: The 18-bit checksum of the components which has been built up in PUTTAG is now searched for in a hash table of length 1024 named TAGPS. Each TAGPS entry contains the 18-bit checksum and a pointer to the entry in TAGS containing the tag string. When a checksum is generated, an entry is looked for in TAGPS (by hashing technique). If no entry is found, one is inserted, and the tag is moved into the next free space in TAGS. If an entry is found in TAGPS, the tag pointed to by it is compared word for word with the tag in TEMPTG. Usually these will be the same, for it is a coincidence indeed if two different tags have the same 18-bit hash sum, but the check is made anyhow. If the tag is already in the TAGS table, then the TEMPTG entries are not copied. If the tag is found not to be in TAGS, it is moved in, and a TAGPS entry is created.

Deck:	HASHTG
Routine:	CLEAR
Type:	Configuration-independent Entry point
Calling sequence:	TSX CLEAR, 4
Function:	Clears the TAGS table and the TAGPS table.

Deck:	HASHTG
Routine:	CANCEL
Type:	Configuration-independent Entry point
Calling sequence:	TSX CANCEL, 4
Function:	Causes the TEMPTG table, which has held the string of tags generated by calls to PUTTAG since the last call to HASHTG, to be reset -- the next call to PUTTAG will start over by storing into the beginning of TEMPTG, as if the latest calls to PUTTAG had never been issued.
Operation:	Resets the TEMPTG table, zeroes the running hash sum of 18-bit items, turns off the switch used to indicate that no tags have been stored yet.

ROUTINES FOR DECK NAMED HASH

System entry points: HASHPR CLRTEM HASHBC

Deck:	HASH
Routine:	HASHBC
Type:	Configuration-independent Entry point
Calling sequence:	CAL WORD TSX HASHBC, 4 PFX 0 (where PFX is ONE, TWO, THREE, or FOUR -- see

below)

Function: If the word in the AC is stored in a table of BCD constituents, then the word is not stored again -- the address of the BCD constituent is retrieved. Thus, HASHBC converts BCD words to addresses pointing to a place where these BCD words are stored. There are two tables where BCD words are stored: the first, called CONTAB (length 997), is never cleared; the second, called ALTTAB (the alternative table, length 299), is cleared by calling CLRTEM (this is done in the CSA at the start of each sentence). The prefix determines which table is used and whether the value must be already present. ONE means value must be in CONTAB; TWO means value is put in CONTAB if not already there; THREE means if value is not in CONTAB it must be in ALTTAB; FOUR means if not in CONTAB, try ALTTAB, if not there, put it in. AC returns address where constituent is stored, or -0 if PFX was ONE or THREE and symbol was not found.

Operation: To hash a BCD word, turn it into a relative address in a narrower range than the totality of BCD words by applying a "hashing function". In this case, the hashing function consists in taking the remainder after dividing the BCD word by 997, and is hence a relative address in CONTAB. We then examine this location in CONTAB -- if it is empty, we know the symbol was never stored in CONTAB, for if it were it would be put in the first empty location; if it is not empty, its contents are compared with the symbol. If a complete match occurs, then we have found the symbol and return the absolute address. If the symbols are different, we must examine the next location of the table and test for equality or emptiness. An ideal size for a hash table is roughly twice the size of the number of items to be stored in it. Under these conditions, the expected number of comparisons needed before finding the address for a BCD word is of the order of 1. 2.

Deck: HASH

Routine: CLRTEM

Type: Configuration-independent
Entry point

Calling sequence: TSX CLRTEM, 4

Function: Clears the table ALTTAB used by HASHBC to store all BCD words not already stored in CONTAB. The CSA calls this routine at the start of the read-in of each sentence.

Deck: HASH
Routine: HASHPR
Type: Configuration-independent
Entry point
Calling sequence:
```
       CAL   PAIR
       TSX   HASHPR, 4
PAIR   PZE   part-of-speech address, , part-of-speech addr.
```
Function: If there is a RULTAB entry for this pair of parts of speech, the AC sign will be set "+". If there is no such entry, the sign will be set "-". Index register 2 is set to the complement of the relative address of where the rule entry is stored, if it is stored, and where it should be stored if it is not stored; i.e., CAL RULTAB, 2 will fetch the first word of the RULTAB entry for the given pair.
Operation: HASHPR uses a hash addressing scheme similar to that used by HASHBC except that each relative address is a multiple of 4, since RULTAB contains 4-word entries.

ROUTINE FOR DECK NAMED PRBCD

System entry point: PRBCD

Deck: PRBCD
Routine: PRBCD
Type: Configuration-independent
Entry point
Calling sequence:
```
TSX   PRBCD, 4
PZE   STACK
```
Function: Prints a STACK read by SENTRD. Inserts blanks around break characters, except that blanks are suppressed before /, comma, and period.

ROUTINES FOR DECK NAMED PRTAG

System entry points: PRTAG PRWORD REGION

Deck: PRTAG
Routine: PRTAG

Type: Configuration-independent
Entry point

Calling sequence:

```
                 CAL   CONST
                 TSX   PRTAG, 4
        CONST    PZE   part-of-speech, , tags
```

Function: Prints a line containing the part of speech followed by the data tags for a given data constituent. The line prints the data tags as they would have appeared on an input card for those data tags (except that the null value for an attribute is printed blank instead of *).

Operation: Scans the string of 18-bit sections forming the data tag, recognizing the beginning of new attributes and values, and separating attribute from value by /, the values by commas, and the separate tags by blanks. Uses PRWORD to print the non-blank portions of the symbols.

Deck: PRTAG
Routine: PRWORD
Type: Configuration-independent
Entry point

Calling sequence:

```
                 CAL   SYMBOL
                 TSX   PRWORD, 4
```

Function: SYMBOL is a BCD character string with trailing blanke. PRWORD puts out the significant BCD characters without the trailing blanks onto the print line. (It calls EDIT to lay out this print line.)

ROUTINE FOR DECK NAMED <u>PRRULE</u>

System entry point: PRRULE

Deck: PRRULE
Routine: PRRULE
Type: Configuration-independent
Entry point

Calling sequence:

```
                 CAL   ADDRESS
                 TSX   PRRULE, 4
                 PZE   INDENT
                 PZE   RTCON, , LFTCON
      ADDRESS    PZE   address of start of subrule
```

Function: To print out the subrule pointed to in the AC whose part-of-speech names (which do not appear in the tags) are RTCON and LFTCON. Prints left half of

subrule on current line, and right half of subrule on next line after indenting INDENT number of spaces. Returns the original AC input.

ROUTINES FOR DECK NAMED DUMP

System entry points: TDUMP DWORD DRULE DBIG DPROG RULSW

Deck: DUMP
Command entry point: TDUMP
Type: CSA monitor-dependent
Configuration-independent
Command entry
Command card format: $ DUMP parameter-1 parameter-2 . . . parameter-n where parameters are any of: WORDS, RULTAB, PROGRAM, BIGTAB.
Function: Causes dump of specified memory areas used by CSA system.
Operation: For each parameter, calls one of the associated subroutines DWORD, DRULE, DBIG, and DPROG.

Deck: DUMP
Routine: DBIG
Type: Configuration-independent
Entry point
Calling sequence: TSX DBIG, 2
Function: Lists the BIGTAB table resulting from syntactic analysis of a sentence. For internal format of BIGTAB, see description of tables in COMMON area at the end of this manual. Before listing BIGTAB, it prints sentence number and date line.

Deck: DUMP
Routine: DPROG
Type: Configuration-independent
Entry point
Calling sequence: TSX DPROG, 2
Function: Dump all of core from location 0 to the entry point of a supplied program called END.

Deck: DUMP
Routine: DRULE
Type: Configuration-independent
Entry point

Calling sequence: TSX DRULE, 4

Function: Lists all rules in the rule table grouped in packets of subrules. If ERRSW is on (meaning grammar compilation unsuccessful), there is no printout.

Operation: Scans through the hashed RULTAB. For all non-empty entries, discovers number of subrules in each rule, and address of first subrule. Since each subrule points to the next consecutive subrule, can print each subrule within a particular rule grouping It prints these subrules by calling routine PRRULE. For format of RULTAB, see description of tables in COMMON area at the end of this manual.

ROUTINES FOR DECK NAMED PRANS

System entry points: PRANS STATIS

Deck: PRANS
Routine: PRANS
Type: Configuration-independent
Entry point

Calling sequence:
TSX PRANS, 4
PZE Complement of first empty space in BIGTAB
PZE Complement of twice number of last word in sentence
PZE Address of first sentence symbol, , number of such

Function: PRANS will print all constituents which span from the first to the last word of the sentence, and whose part of speech is a user-defined "sentence symbol" (e.g., "S" or "PRED"). PRANS prints these in tree form, showing each node with all its tags, the line in BIGTAB corresponding to it, and the level number on the tree. In addition, for each tree printed, PRANS will print a list of all values appearing on the tag "R" or "RULE" attribute for a node. If option TNODE is specified, all nodes which were suppressed in execution because they were identical in span and in tags to another are printed out with their constituent trees.

Operation: When PRANS finds the top of a tree to be printed out, it scans down the tree. It saves the not yet printed branch address for level n in the nth level of a push-down list, PUSH1. To print a branch, it calls MODUL.

Deck: PRANS
Routine: MODUL
Type: Configuration-independent
Internal to PRANS
Calling sequence: TSX MODUL, 7
(Index register 2 is the complement of the relative address in BIGTAB of the node to be printed.)
Operation: Used with the generator program to generate nodes. MODUL prints the node at BIGTAB + 2, 2 (third word in entry, pointing to part of speech and tags) using PRTAG, after it has indented a number of indentation units equal to the level count. At the position where constituents for level n are printed out, MODUL prints the character "I" for all levels where a node is not yet printed out, giving the character of a tree with vertical lines linking pairs of nodes on the same level.

ROUTINE FOR DECK NAMED DATE

System entry point: DATE

Deck: DATE
Routine: DATE
Type: Configuration-independent
Entry point
Calling sequence: TSX DATE, 4
Function: Prints sentence number and date information of the latest $DATE card in one line. Uses PRBCD for printing date information from the monitor date stack.

ROUTINES FOR DECK NAMED GRAMAR

System entry points: GRAMAR ERRSW

Deck: GRAMAR
Command entry point: GRAMAR
Type: CSA monitor-dependent
Configuration-independent
Command entry point
Command card format: $ GRAMMAR
Function: Reads rule packets following $ GRAMMAR card until another dollar sign card or end of file is reached.

Compiles these rule packets into entries in RULTAB (see description of tables in COMMON area) pointing to subrules stored internally in TAGTAB. Lists any formation errors in these rules, and turns on the switch ERRSW (an entry point) if there are any errors.

Operation: Reads * card serving as a rule header, and calls HASHPR to get a RULTAB entry for a pair of parts of speech. All symbols are first converted to 15-bit addresses by calling HASHBC (specifying the "permanent" hash table). (See "HASHBC".) Reads subrule cards, and inserts the compiled tag conditions into the table TAGTAB. Uses subroutines NEXT and PUT (internal routines) to fetch constituents from input cards and to store generated rule tags.

Deck: GRAMAP

Routine: PUT, PUTAB

Type: Configuration-independent
Internal to GRAMAR

Calling sequence:

```
        CAL    TAG
        TSX    PUT, 4
TAG     BCI    1, SYMBOL
```

Function: PUT converts the symbol in the AC to a 15-bit address. It then inserts 3 additional bits: Bit 18 if switch ATTR is on, bit 19 if NEWCN is on, and bit 20 if switch MINUS is on. It resets the first two of these switches after every call. The 18-bit quantity then obtained is inserted into the next free position in the TAGTAB stored in GRAMAR to hold subrule tags. Where the conversion to a 15-bit address must be avoided, i.e., for symbols 1 and 2, calls PUTAB instead of PUT.

Operation: PUT first calls HASHBC to get a 15-bit address. PUTAB skips this step. Then the additional bits are inserted, the switches reset, and the result stored in the next free TAGTAB location.

Deck: GRAMAR

Routine: NEXT

Type: Configuration-independent
Internal to GRAMAR

Calling sequence:

```
TSX    NEXT, 4
...exit if character in "STACK" was =
...exit if character was +
...exit if character was , or /()
...exit if character was /
```

. . . exit if character was anything else.

Function: Gets next item in "STACK" resulting from reading a subrule, and transfers to the appropriate exit. A "STACK" is defined on the card specifying subroutine SENTRD.

Deck: GRAMAR
Table: TAGTAB
Type: CSA subrule table
Size: 7500
Format of a single subrule:
MZE symbolic label (if any), , address next subrule
PZE success exit, , failure exit
Strings of 18 bits as follows:
XYZ address of attribute or value
X = 1 for start new attribute, 0 for value
Y = 1 for start new constituent (C_1, C_2, or C_3)
Z = 1 for negative sign (all constants following X- on a tag condition)
Numbers on rules (e.g., ETC/2 or ATT/2-X) are stored as absolute values.
A fence of 077777 terminates a subrule.
If a constituent is tagless, then 600000 is present to indicate new constituent but zero attribute.
/* appears as half word 000000.

ROUTINE FOR DECK NAMED SENTNC

System entry points: SENTNC TAPSW COUNTS

Deck: SENTNC
Command entry point: SENTNC
Type: Monitor-dependent
Configuration-independent (except for reference to SNTAP)
Command entry point
Command card format: $SENTENCES
Function: Reads words with their constituent tags from input (or from A5 if option LOOKUP was specified). Counts sentence and word numbers. When a sentence is read in, the syntactic alternatives are stored in WORDT. For each word, an entry in WORDS is created pointing to where in WORDT the first alternative constituent for that word is, and telling how many alternatives there are. At the end of a sentence (** card) control

is returned to the routine which last called monitor (which is the main program for syntactic analysis). This routine (deck name ANALYZ) will parse the sentences and return directly to entry point SENTNC.

DECK NAMED ANALYZ

Main program. No entry points.

Deck: ANALYZ
Main program

Type: Configuration-independent
Calls monitor

Function: Calls monitor first to process all grammar reading, setting of options, and other functions performed by commands under the monitor. When the SENTENCES card is reached, the SENTNC program will return control to this main program at the end of each sentence. The ANALYZ deck will parse the sentence whose constituents are defined in the WORDS table, creating a BIGTAB, showing all the legitimate constituents of the sentence. It does this by combining every pair of adjacent constituents according to the grammar into a new constituent, by calling GETRL with every pair of adjacent constituents. GETRL will return the list of valid new constituents, and the process of combination is continued until all possibilities have been tried. Then PRANS is called to print the results out and control is returned to SENTNC for the next sentence to be parsed.

ROUTINES FOR DECK NAMED GETRL

Entry points: GETRL DOLL

Deck: GETRL
Routine: GETRL
Type: Configuration-independent
Needs CSA grammar RULTAB in COMMON
Entry point

Calling sequence:

```
          CAL    C1
          LDQ    C2
          TSX    GETRL, 4
C1        PZE    part of speech, , tags (for left constituent)
```

```
G2          PZE   part of speech, , tags (for right constituent)
```

Function; GETRL returns in the AC:

PZE address of first created constituent, , number of created constituents (or zero if none). GETRL finds the rule packet, if any, for the part-of-speech pair read into it. If there is a rule packet, GETRL starts to apply the first subrule. After applying a subrule, GETRL examines the transfer address of the subrule. If the subrule succeeded, the next subrule is taken from the success transfer. If the subrule failed, the next subrule is taken from the failure transfer. If the appropriate transfer address is zero, GETRL is through with the given rule packet. It then returns the pointer to the list of new constituents which it has generated for successful subrules (if any).

Deck: GETRL
Routine: GETATT
Type: Configuration-independent
Internal to GETRL

Calling sequence:

```
            CAL   attribute
            LDQ   constituent
            TSX   GETATT, 4
            . . . return if attribute not found. . .
            . . . return if attribute found. . .
attribute   PZE   address of attribute
constituent PZE   , , address of data tag to be searched:
                  TAGTAB in HASHTG
```

Function: Scans data tag looking for an attribute which matches the given attribute. If not found, returns 1, 4. If found, returns 2, 4, with the ability to call GETVAL to get the values of this attribute.

Deck: GETRL
Routine: GETVAL
Type: Configuration-independent
Internal to GETRL

Calling sequence:

```
            TSX   GETVAL, 4
            . . . return if exhausted all values. .
            . . . return if routine is supplying a value. . .
```

Function: This routine is called after GETATT has been called and has found an attribute on a specific data constituent. Each call to GETVAL gets the next value on the attribute. If there are no more values, exits 1, 4. If there are values, exits 2, 4 with value in AC.

ROUTINE FOR DECK NAMED OPTION

System entry points: OPT SHORT TNODE NOMAX LOOKUP RULTAP

Deck: OPTION
Command entry point: OPT
Type: Monitor-dependent
Configuration-independent
Command entry point
Command card format: $OPTION parameter-1 parameter-2
Function: Acceptable parameters are: RULTAP TNODE LOOKUP NOMAX SHORT. When any of these parameters is encountered, the switch with the same name is turned on. All unspecified switches are turned off. These switches are system entry points which may be interrogated by other commands or subroutines in the system. Future options may be included by expanding the list of parameters compared and expanding the vocabulary of entry points.

DECK NAMED END

System entry point: END

Must be present to tell routine DPROG in DUMP where the end of the CSA program is.

TABLES IN COMMON AREA: BIGTAB RULTAB WORDS

Table: BIGTAB
Location: COMMON
Size: 5000 words (1666 entries)
Each entry format:
PZE right word number,, left word number
PZE right subconstituent address,, left subconstituent address
XY0 part-of-speech address,, address of tags
(X = 1 if this entry has the same span and constituent form as an earlier one)
(Y = 1 if another entry later in the table is marked X = 1 by virtue of being identical in span and constituent to this entry, and X = 0 for this entry.)
Word numbers are stored in the form of the 2's

complement of twice the number, e.g., word 15 would be the 2's complement of 30 (octal 77742). The subconstituent addresses are the 2's complements of the relative address in BIGTAB of the entry being referred to.

Table:	RULTAB
Location:	COMMON
Size:	62 four-word entries
Each entry format:	PZE C_1,, C_2
	PZE number of accesses,, number of failures
	PZE address of first subrule,, number subrules
	PZE NOT USED (should be dropped entirely)

C_1 contains left part of speech
C_2 contains right part of speech
The subrules are stored in a table called TAGTAB, internal to the deck GRAMAR. The format of this table is shown in description of the GRAMAR routine.

Table: WORDS

(See description of deck SENTNC.)

APPENDIX I-B: Dictionary Assembly/Update Program Logic Manual

Table of Contents

ROUTINE FOR DECK NAMED <u>MONITOR</u>

Deck: MONITOR
Routine: MONITOR
Type: Transfer vector
No entry point
Configuration-independent
Function: Must be present to transfer the machine to the assembly and updating system. Practically a start card. Currently transfers to PARAMS. May later be extended to a full-fledged monitor.

ROUTINES FOR DECK NAMED <u>CONTROL</u>

System entry points: DICTPR INTAPE OUTPUT DUMMY
OUTAPE CDTAPE CARDPR DATE RUSS

Deck: CONTROL
Routine: PARAMS
Type: Configuration-independent
Update system entry point: PARAMS
Function: This program reads the parameters in a STACK whose contents are provided by SENTRD. (Uses READP deck by TSX SENTRD, 4.) Checks for correct parameters and prints error messages. If parameters are incorrectly specified, program stops to allow for corrections according to the printout message. Undefined parameters are set by default. Acceptable parameters of the $UPDATE card, which cause a setting of the respective switches, are: RUSSIA, PRCARD, NOCAPR, PRDICT, NODIPR, NOIN, NOOUT, UPDIN, UPDOU, DATE, DEBUG. The switches are system entry points which may be interrogated by other commands or subroutines in the system. Additional parameters can be included by expanding both the list of acceptable parameters and the vocabulary o. entry points.

Deck: CONTROL
Routine: SORTQ
Type: Configuration-independent
Internal to CONTROL
Direct transfer point
Function: If assembled update records are to be test-printed

because parameter DEBUG was specified, SORTQ writes them all out in octal representation and quits. If DEBUG was not specified, SORTQ defines the sort area, sort/merge tapes, and buffer areas for the assembled update records. These sort parameters are interrogated by the Sort/Merge system.

ROUTINES FOR DECK NAMED <u>ASSLY</u>

System entry points: INPUT SORT SORTM

Deck: ASSLY
Routine: INPUT
Type: Configuration-independent
Entry point
Calling sequence: TSX INPUT, 4
Function: Assembles dictionary update records. The Sort/Merge system will call INPUT to get an assembled dictionary record to sort on. Before INPUT branches back to its caller, the accumulator must contain the address and the word count of the assembled record. If the accumulator contains + Ø, the input record end-of-file has been reached INPUT interrogates the language parameter to see whether Russian words are to be processed. If so, all "Cyrillic" BCD words are converted into an internal dictionary sort code to collate on Cyrillic sequence rather than 7094 standard 9-code. This routine reads "logical cards" of adds and deletes for the dictionary. A logical card runs from columns 1-72, unless column 72 is punched with an 11-punch (minus sign), in which case it includes columns 1-71 plus continuation from the next physical card. Any number of physical cards can form a logical card as long as a "-" in column 72 indicates that the next card is a continuation. In a logical add card, an * in column 1 indicates that the card contains a natural language word that will constitute the argument field in the corresponding assembled record. Absence of an * in column 1 of a logical add card indicates that the card contains a syntactic alternative which will become the function field in the assembled record. Add cards are preceded by a $ADD control card. In a logical delete card, an * must appear in column 1 followed by the word to be deleted. If

certain alternatives only are to be deleted, their numbers (separated by commas) must follow the word and be separated from it by at least one blank. Delete cards are preceded by a $DELETE control card. All assembled update records contain in their first 36-bit word the information ADD or DEL and the alternative number(s) (Ø for master record). Assembled records consist of an ADD or DELETE control word with alternative number(s) (Ø for master), a date-word, an argument field (which is converted into Cyrillic collating sequence if it is Russian), a 36-bit zero word, and a function field (followed by another 36-bit zero word if it is an add entry). A $END card closes the assembly. If an invalid update control is found, an error message is printed out and the job is interrupted.

Deck: ASSLY
Routine: SORT
Type: Configuration-independent
Entry point
Calling sequence: TSX SORT, 4
Function: This routine compares two records to tell the Sort/Merge system in which sequence each record is to be written out. Index register 1 and index register 2 contain the pointers to the two records, i.e., CAL 1, 1 will get the first word of record 'A' and CAL 1, 2 will get the first word of record 'B'. It must return 3, 4 if record A is greater than B, low-to-high sort, 2, 4 if record A equals B low-to-high sort, and 1, 4 if record A is less than B, low-to-high sort. It must return 1, 4 if record A is greater than B, high-to-low sort, 2, 4 if record A equals B, high-to-low sort, and 3, 4 if record A is less than B, high-to-low sort. Index registers 1 and 2 must not be destroyed before returning. In the update procedure, this routine performs a comparison such that the assembled dictionary records are sorted on the natural language word, low-to-high, as major field, on the alternative number, low-to-high, as intermediate field, and on the delete or add control, high-to-low, as minor field.

Deck: ASSLY
Routine: SORTM
Type: Configuration-independent
Entry point

Calling sequence: TSX SORTM, 4

Function: This routine is identical to SORT, but is only transferred to during the final merge. All return commands are the same as in SORT. SORTM is not used if the dataset is to be sorted only once.

ROUTINE FOR DECK NAMED UPDATE

System entry point: UPDATE

Deck: UPDATE
Routine: UPDATE
Type: Configuration-independent
Entry point
Direct transfer point

Function: This routine is the heart of the updating program. It calls in assembled, sorted logical update records and logical master dictionary records, matches them, deletes logical records, adds logical records, and updates the dictionary file. After reading in a dictionary record, UPDATE reads in an assembled record and compares it with the dictionary record. If the dictionary record is lower than the assembled record, the matching dictionary record has not yet been reached and the dictionary record is written unchanged on the updated file. If the dictionary record is higher than the assembled record, the latter is a completely new entry and is written in its entirety on the updated file. If the dictionary record is equal to the assembled record, a test for DELETE or ADD is made. If the assembled record is to delete the dictionary entry, the next dictionary record is read in, thereby erasing the entry to be deleted. If the assembled record is to be added to the dictionary entry, it is a new alternative and the dictionary entry is first written with all alternatives unchanged on the updated file. Only then is the new alternative record written on the file, immediately following the last old alternative. If a record is to be changed, it is actually deleted, and then replaced by a newly added record.

ROUTINES FOR DECK NAMED DICPNT

System entry point: DCTPN

Deck: DICPNT
Routine: DCTPN
Type: Configuration-independent
Entry point
Calling sequence: CLA IOWORD
TSX DCTPN, 4
Function: Writes BCD dictionary records on a print file. Index register 1 must contain the file name of the BCD dictionary printout file before this calling sequence is executed. The file name is a decimal digit 0, 1, 2, etc. to be used by IOFMS, a set of input-output tape control routines not described in detail here. IOWORD contains the address of the logical record to be printed in its address field and the number of words in this record in its decrement field. If the language parameter RUSSIA was specified in the $UPDATE card, a conversion from the internal "Cyrillic" character collating sequence into the standard 7094 BCD collating code will take place. DCTPN composes a print line which is the BCD representation of a logical dictionary record and sends it to PPNT in IOFMS which writes it out in blocked format. The BCD dictionary print file should be printed on the 1401 by the 1401 PRINT PPNT PROGRAM, but can also be printed less efficiently via FMSII.

Deck: DICPNT
Routine: STORC
Type: Configuration-independent
Internal to DCTPN
Calling sequence: TSX STORC, 4
Function: Stores BCD characters one by one in a 20-word (120 character) print line. Accumulator must contain the character right-adjusted before one can transfer to this routine.

Deck: DICPNT
Routine: PPNOT
Type: Configuration-independent
Internal to DCTPN
Calling sequence: TSX PPNOT, 4
Function: Dumps completed print-line buffer of 120 characters.

ROUTINE FOR DECK NAMED **READP**

System entry point: SENTRD

Deck: READP

Routine: SENTRD

Type: Configuration-independent
Entry point

Calling sequence:
```
TSX    SENTRD, 4
PZE    STACK, , SIZE
```

Function: The user can specify a STACK of any size he wishes. SENTRD will read a logical card as defined above. SENTRD looks for strings of consecutive alphabetic characters, separated by blanks or by "break characters" (comma, dollar sign, period, parentheses, slash, equals sign). Each string of alphabetic characters between breaks or blanks, as well as each individual break character, is stored, left justified, padded with blanks, in the next free location in STACK. If a string is longer than 6 characters, the components of the string are separated by a "logical concatenator" symbol consisting of the word 767676767676. Blanks on cards, which are used only to delimit fields of text characters, are not put into the STACK. After the last word has been stored in STACK, a fence of 777777777777 is stored. SIZE is the maximum number of words (including fence) which will be stored in the STACK. Overflow will be lost, and the fence stored in the last location, if the logical card has too many fields.

In the Assembly/Update system the task of this routine, which is also utilized in the Combinatorial Syntactic Analyzer, is to stack and edit the update parameters.

Example: Suppose a logical update control card contains:
$UPDATE ENGLISH, NOCAPR, PRDICT, UPDIN, UPDOU, DATE 090966

The stack will contain:

$
UPDATE
ENGLIS
767676767676
H
,
NOCAPR
,

PRDICT

,

UPDIN

,

UPDOU

,

DATE
090966

The first 72 characters of this update card will also be printed for a visual check.

ROUTINE FOR DECK NAMED CARDRD

System entry points:	CARDRD REDSW
Deck:	CARDRD
Routine:	CARDRD
Type:	Update-system independent I/O tape dependent Entry point
Calling sequence:	TSX CARDRD, 4 PZE USERBF, , LENGTH . . . end-of-file return. . . . . . redundancy return. . . . . . normal return. . .
Function:	Reads one physical card image into user's buffer from an input tape (whose number may be redefined at any time). Currently the 7094 card input tape A2 is used.

ROUTINES FOR DECK NAMED INTER

System entry points:	EDITX BCDCAN PHYREW WTCHA WTCHB CHKAG CHKBG WA3 MEGFIL GETREC PUTREC OPENIN OPENOU CLOSOU
Deck:	INTER
Routine:	EDITX
Type:	7094 tape I/O dependent Entry point
Calling sequence:	TSX EDITX, 4 . . . EDIT commands (see IBM Research Computing Center edit manual). . .

Function: Checks whether Channel A is still in operation when the user wants to use it for EDIT. Since, in addition to EDIT, IOFMS, Sort/Merge, and CARDRD also use Channel A, their Channel-A operation must be completed before EDIT can use it. EDITX serves to finish up such an operation, if it is in progress. It also saves the redundancy status of Channel A for the card read routine by setting a switch which is tested in CARDRD.

Deck: INTER
Routine: MEGFIL
Type: 7094 tape I/O dependent
Entry point
Calling sequence:

```
TSX    MEGFIL, 4
PZE    USERBF, , LENGTH
```

Function: Will get one logical record from the merge file and put it into the user's buffer. The address of the user's buffer and its LENGTH must be specified in the control word following the TSX instruction. Uses TSX SORTX, 4 to get the next following logical record from either one of two files to be finally merged.

Deck: INTER
Routine: GETREC
Type: 7094 tape I/O dependent
Entry point
Calling sequence:

```
TSX    GETREC, 4
PZE    USERBF, , LENGTH
```

Function: Will get one logical record from an input file and put it into the user's buffer. The address of the user's buffer and its LENGTH must be specified in the control word following the TSX instruction. Uses TSX READ, 4 to get a logical input record.

Deck: INTER
Routine: PUTREC
Type: 7094 tape I/O dependent
Entry point
Calling sequence:

```
TSX    PUTREC, 4
PZE    USERBF , LENGTH
```

Function: Will take one logical record as defined by the control word following the TSX instruction and put it on the output file. Uses TSX WRITE, 4 to output the logical record. If the parameter PRDICT appears on the $UPDATE card, PUTREC also gives the dictionary

BCD print routine the location of the logical record, so that the latter can put it (in BCD form) on a print file.

Deck: INTER
Routine: OPENIN
Type: 7094 tape I/O dependent
Entry point
Calling sequence: CAL INTAPE
TSX OPENIN, 4
Function: Tells the input/output tape control routine IOFMS which input file is to be opened. The name of the file must be in the accumulator before one can TSX to this routine. In the Assembly/Update system, the input master dictionary file has to be opened.

Deck: INTER
Routine: OPENOU
Type: 7094 tape I/O dependent
Entry point
Calling sequence: CAL OUTAPE
TSX OPENOU, 4
Function: Tells IOFMS which output file is to be opened. The name of the file must be in the accumulator before one can TSX to this routine. In the Assembly/Update system, the output master dictionary file and, if parameter PRDICT (print dictionary) was stated, the output BCD print dictionary file have to be opened.

Deck: INTER
Routine: BCDCAN
Type: 7094 tape I/O dependent
Entry point
Calling sequence: CAL file name
TSX BCDCAN, 4
Function: Returns in the accumulator the 15-bit internal address of the device possessing the given file name. File name must be in the accumulator before one can TSX to this routine. Only used for compatibility with CSA system. In the Assembly/Update system, BCDCAN transfers immediately back to caller. See also BCDCAN routine in CSA program logic manual.

Deck: INTER
Routine: PHYREW
Type: 7094 tape I/O dependent

Entry point
Calling sequence: CAL file name
TSX PHYREW, 4
Function: Rewinds a tape with a file whose name is in the accumulator at the time this routine is entered. Only used for compatibility with CSA system. In the update routine, PHYREW transfers immediately back to caller. Rewinding operations are initialized by the routine CLOSOU (which see).

Deck: INTER
Routine: CLOSOU
Type: 7094 tape I/O dependent
Entry point
Calling sequence: CAL file name
TSX CLOSOU, 4
Function: Closes all output operations, initializes rewinding of tapes and terminates I/O subroutines. Uses ENDIO of IOFMS. Logs all record counts (logical and block records, all frame numbers, total record counts of all files).

Deck: INTER
Routine: WTCHA
Type: 7094 tape I/O dependent
Entry point
Calling sequence: TSX WTCHA, 4
Function: Makes Channel A available to the input/output tape control routines IOFMS. Called normally from the I/O and transfers to the Sort/Merge system.

Deck: INTER
Routine: WTCHB
Type: 7094 tape I/O dependent
Entry point
Calling sequence: TSX WTCHB, 4
Function: Makes Channel B available to IOFMS. Called normally from I/O and transfers to the sorting routine.

Deck: INTER
Routine: CHKAG
Type: 7094 tape I/O dependent
Entry point
Calling sequence: CHKAG, 4
Function: [illegible] Channel A available to the Sort/Merge system. Called normally from the Sort/Merge system and

transfers to IOFMS.

Deck: INTER
Routine: CHKBG
Type: 7094 tape I/O dependent
Entry point
Calling sequence: TSX CHKBG
Function: Makes Channel B available to the Sort/Merge system. Called normally from the Sort/Merge system and transfers to IOFMS.

Deck: INTER
Routine: WA3
Type: 7094 tape I/O dependent
Entry point
Calling sequence: TSX WA3
Function: Is a request by the Sort/Merge system to write out messages. Transfers to the write routine W 4ø3 of IOFMS.

II. THE CSA RUSSIAN GRAMMAR: LINGUISTIC RESEARCH AND RELATED LANGUAGE PROCESSIN'G ACTIVITIES

Alexander Andreyewsky

II. THE CSA RUSSIAN GRAMMAR: LINGUISTIC RESEARCH AND RELATED LANGUAGE PROCESSING ACTIVITIES

2.0 Introduction

This section of the report describes that portion of contract-supported linguistic research and related language processing which was directed primarily towards the development of a Russian surface grammar for the CSA system. The main results of the linguistic research on Russian grammar, which involved a cyclical process of formulation, testing, and revision of parsing rules in CSA format, are presented in Section 2.1. Section 2.2 describes an independent exploratory study on subclassification of Russian parts of speech which was conducted with the assistance of Library of Congress personnel. Related language processing activities, consisting principally of Russian Master Dictionary (RMD) updates and processing of a large Russian text corpus preliminary to obtaining statistical information, are summarized in Section 2.3.

2.1 Grammatical Research on Russian for CSA

The central focus of the linguistic research conducted under the contract was a study of Russian surface structure phenomena leading to the development and testing of grammar rules for parsing Russian sentences. The results of this study are reflected in part in two experimental Russian grammars, RG1 and RG2, expressed in the tag metalanguage of the CSA system (cf. Section 1.2). RG1, a relatively small grammar employed largely for system debugging purposes, was based on only the first 500 words (20 sentences) of a 160-sentence sample drawn randomly from a 30,000 word corpus (1600 sentences) of Pravda editorials. Since RG1 has been entirely superseded by RG2, its considerably more extensive successor, only the latter is described in this report.

The second experimental grammar, RG2, based on both the entire 160-sentence Pravda sample and a variety of referencer on Russian grammar, constitutes a relatively extensive preliminary set of grammar rules for surface structure recognition of Russian sentences. Its formulation was regarded as one stage in a cyclical process consisting of formulation, testing, and revision of grammar rules. Consistent with this approach, a number of recurrent linguistic problems that were encountered in testing the rules of RG2 were subsequently investigated in greater detail, an activity which in a number of instances led to the formulation of new rules or the revision of existing ones. The present section includes a discussion of a representative sample of RG2 rules grouped according to major grammatical topics. Where applicable, the discussion of a given topic includes a summary of the findings of subsequent linguistic investigations and present any sets of new or revised rules that have been tentatively proposed as a result.

2.1.1 Definitions and Underlying Assumptions

Since Soviet works on Russian grammar were referred to extensively in the linguistic research on the CSA Russian grammar, a number of terms and concepts employed in those sources have been adopted in this section of the report. Because of the likelihood that they will be unfamiliar to many readers, explanations of concepts and definitions of terms are provided here, along with a statement of certain assumptions involved in the present study.

As described in Section 1.1 of this report, prior to analysis each "word" in a sentence, including sentential punctuation,[1] is assigned a set of mutually exclusive syntactic alternatives, each of which is in the form of a structured symbol consisting of a part-of-speech name[2] followed by a (possibly null) string of tags. A complete analysis of a sentence is one in which systematic combination of adjacent syntactic alternatives and higher-order constituents according to the rules of a grammar results in formation of a sentence constituent S which spans the entire input sentence. In the CSA Russian grammar, it is assumed that each such S immediately dominates a constituent PRED, together with any associated sentential punctuation. The PRED constituent, composed of the subject and predicate of the sentence, is referred to in this report as the predication constituent or predication. All other constituents are collectively referred to as non-predicative constituents.

In discussing various members of the set of part-of-speech classes, the distinction commonly drawn between lexical and function[3] words will occasionally be employed. In addition, an analogous distinction will be made between multi-word constituents that can function by themselves as higher-order constructions and those that can function only as constituents of higher-order constructions. The necessity of defining constituents of the latter type, which will be called quasi-constituents, is primarily a by-product of the binary branching format required by the current version of the metalanguage. For example, in parsing a coordinative construction like "Peter, John and Mary" it is necessary at some point to create the quasi-constituents ", John" and "and Mary" which are not higher-order constructions in a linguistic sense but combine with the constituent "Peter" to form one.

Two lexical words (or any two non-predicative constituents) linked by predicative agreement (i.e., subject-predicate agreement) form a predication; if they stand in apposition to one another, they constitute an apposition[4]; if linked by non-predicative agreement, government, or adjoinment (primykanie), they form endocentric word groups (slovosochetanie[5]).

Constituents not linked to any other constituent in one of the ways just enumerated (sometimes referred to as "parenthetic" constituents[6]) are said in the Soviet literature to be unrelated to the surface structure of the sentence.

Lexical words, predications, appositions, and word groups can be conjoined by coordinative and subordinative conjunctions, in which case the resultant constituents are referred to as *compound* and *complex*, respectively.[7] In certain situations, strings of adjectives, nouns or numerals whose members are identical in syntactic function can form *accumulative strings*.[8] For instance, in BOL6WO1 BELY1 KAMENNY1 DOM ('a large white stone house') the adjectives form such a string.

Two function words can produce only a quasi-constituent. However, there are three basic possibilities for combining a function word with a lexical word, a predication, an apposition, or an endocentric word group:

1. a preposition combines with its governed complement to form a prepositional phrase whose name is additionally qualified by the class of the complement (e.g., preposition-ordinal phrase, preposition-noun phrase);
2. conjunctions combine with predications to produce *clauses* identified by the class of conjunction as either *coordinate* or *subordinate*;
3. conjunctions in combination with non-predicative constituents form coordinate or subordinate *phrases*. (The term *phrase* is also employed in referring to accumulative strings and is used interchangeably with the term *word group*.)

Subordinate clauses and phrases are typically set off by paired commas; coordinate clauses and phrases, where appropriate, are separated by commas and other punctuation. Constituents of predications and of word groups linked by government cannot be separated or set off by punctuation. Appositions and word groups linked by agreement or adjoinment are typically not punctuated. However, dependent constituents in these constructions can be set off by commas or equivalent detaching punctuation from the dominant constituent in the respective group. This circumstance is noted by referring to the constituent set off by commas or equivalent punctuation from the dominant word as being *detached*. The following contains a detached participial phrase: STOL, POKRYTY1 GAZETO1, ... ('a table covered with a newspaper ... ').

Constructions and constituents which are formed by predicative ties, appositive ties, government, agreement, or adjoinment and which do not contain[9] any punctuation or conjunctions are referred to as *simple*; all others as *complicated* (*oslozhnennye*).

2.1.2 Restrictions on the Scope of RG2

In addition to being confined to surface structure phenomena, restrictions of two other kinds were imposed on the scope of RG2: (a) restrictions on the range of construction types covered and (b), within certain of the types considered, restrictions on the nature of coverage as a result of various simplifying assumptions. The principal restrictions in the former category were the following: The only sentence types considered were declarative sentences containing neither direct quotes nor strings of characters of formal or natural languages other than Russian.[10] The main focus of coverage in RG2 is placed on simple word groups and predication constituents. A limited capacity for recognizing appositions and compounds was also included. Subordinate clauses, generally limited to relative KOTORY1-clauses and to CTO- and CTOBY-clauses, were dealt with only on a token basis. Recognition of a limited number of detached constituent types was also attempted, but proved to be a difficult task because of the need for additional work on recognition of associated punctuation (mainly separating and detaching commas). In the course of testing the grammar on the 160-sentence Pravda editorials sample, a few ad hoc rules were introduced to bridge gaps in the coverage of the grammar. Such temporary ad hoc rules were clearly labeled as such; however, since they are linguistically uninteresting, such rules are not discussed further in this report.

The most significant simplifying assumption involved the treatment of preposition-noun phrases, which were arbitrarily linked to left-adjacent potential governors with the following exceptions. Verbs and predications were allowed to combine with preposition-noun phrases on either side and were given preference in instances where a prepositional phrase had more than one adjacent potential governor. The rules governing combination of adverbs with adjacent constituents were also considerably oversimplified, due to the absence of detailed information on subcategorization and selection restrictions.

Another area where major simplifications were made was that of redundant analyses, i.e., sets of analyses such that the availability of one member is sufficient to predict the nature of the remaining members. Whether such analyses were trivially different[11] (i.e., did not correspond to significantly contrasting meanings) or differed in important respects,[12] single structural descriptions were arbitrarily assigned to some of the more frequent patterns in order to avoid unnecessarily voluminous output.

Word order restrictions -- except for the more obvious instances, such as preposition-noun sequence, adjective-noun sequence, and (cardinal numeral)-noun sequence, as well as the artificial restrictions on the position of preposition-noun phrases relative to their "governors", -- were not imposed in the rules of RG2; nor could the linguistic problems of word order be adequately explored during the contract period.[13] Accordingly, constructions like OTVET $AVSTRII $VENGRII are given two analyses, one of which is correct ('reply of Austria to Hungary'), the other incorrect ('reply of Hungary to Austria').[14]

2.1.3 Employment of the Metalanguage in RG2

In view of the flexibility of the metalanguage (Section 1.2), a number of alternatives are in general available for expressing the same linguistic facts in the form of CSA grammar rules. Before turning to a description of individual rules, two instances where the choice of a particular alternative in RG2 affected broad segments of the grammar will be discussed. The first instance involved two measures designed to reduce the number of tags associated with most constituents:

1. The case and person attributes were combined into a single abstract attribute CP, necessitating a split of the traditional nominative case into three versions -- those of the first, second, and third person;
2. Number and gender were combined in similar fashion into an attribute NG with values plural (P), masculine (M), feminine (F), and neuter (N).

While the decision to combine case and person was not without benefit, in retrospect it would have been better to keep number and gender separate, because in instances where only number agreement is required, duplication of rules results.[15] Moreover, the loss of gender distinctions in the plural is disadvantageous in some situations.[16]

More serious practical difficulties arose in the second instance as a result of the approach chosen for enforcing a particular order of concatenation in endocentric constructions -- that of renaming constituents. The two following equivalent sets of rules for a hypothetical constituent "A" illustrate the major alternatives.

(1) A + A = AP
A + AP = AP

(2) A PHRASE/* + A = A PHRASE/PLUS

Comment: The resultant constituent (C_3) in (1) is called an A-phrase (AP). The C_3 in (2) is an A which has the attribute "phrase" (PHRASE/PLUS). The first constituent (C_1) in (2) can be any A, but the attribute PHRASE must either have no value or not be present at all, as indicated by the *.

In the constituent renaming approach (1) it is assumed that there are no rules of the form AP + A = AP and AP + AP = AP in the grammar. Moreover, not only are two rules necessary to accomplish what can be done in (2) by a single rule, but also almost every rule in which A enters as a constituent must be duplicated to take care of the constituent AP. An analogous problem arises when a constituent V (verb) can govern several constituents N (noun) on either side. Rules of the type V + N = V and N + V = V would produce multiple redundant analyses of a string like N V N N. In order to avoid this, the following equivalent rule types are possible.

(3) V + N = VP
VP + N = VP
N + V = V

(4) V + N = V PHRASE/PLUS
N + V PHRASE/* = V

Comment: In (3) VP stands for some constituent "verb phrase". In (4), the class name of the constituent is not changed and VP is written as V PHRASE/PLUS with a resultant saving in the number of rules in which a verb or verb phrase is either produced or combined with another constituent.

While the duplication of rules is avoided in approaches (2) and (4), the additional tags proliferate so that the rules become extremely long and correspondingly difficult for a human to interpret. The approach illustrated in (1) and (3) was selected for RG2 with an understanding of some of its consequences, but without full knowledge of what their magnitude might be.[17] The device of ordering of subrules (Section 1. 2. 5) was not employed in RG2 because it was written before this feature was added to the metalanguage.

2. 1. 4 The CSA Russian Grammar

In order to present a representative sample of RG2 rules, while at the same time illustrating the results of further investigation of linguistic problems, the description of the CSA Russian grammar is organized according to the major grammatical topics addressed. The description begins (A) with a relatively extensive treatment of the principal types of noun phrases, prepositional phrases, and their components, followed by discussions on the handling of predications (B) and compound constituents (C). It concludes (D) with a brief analysis of problems encountered in the recognition of punctuationally delimited constituents followed (E) by a summary of activity on the lexicon developed in support of RG2.

A. Noun Phrases, Prepositional Phrases, and Their Components

In the case of noun phrases and their components, the discussion will be restricted to constituents resulting from the combination of long-form adjectives, cardinal numerals, and nouns, as well as of constituents resulting from such combinations. Constructions involving the combination of constituents of the same part-of-speech class will be considered first. These include: (1) adjective strings, (2) cardinal numeral strings, and noun-noun constructions, the latter being subdivided into (3) noun-noun predications, (4) close appositions, and (5) word groups formed by a governing noun. The remainder of the discussion is concerned with constructions involving two or more distinct parts of speech, including: (6) nouns with cardinal numerals, (7) nouns with adjectives, (8) nouns with cardinal numerals and adjectives, and (9) preposition-noun phrases.

1. Adjective Strings

Only long-form adjectives of the types shown in Table II-1 are considered here.[18] Single adjectives and unpunctuated strings of adjectives, the accumulative strings, modify a noun successively, i.e., the bracketing is of the general form (A (A (A N))), where A and N stand for adjective and noun, respectively. Since, however, adjectives can form a variety of other strings [19] the accumulative strings are assigned the following type of bracketing ((A (A A)) N) in order to make the treatment of adjectival strings uniform.

Table II-1: Subclasses of Adjectives Distinguished in RG2.[20]

Long-form Adjectives (A)				
Sublasses			Tags Used	Examples
Adjectives proper	Degree of comparison	Positive	SC/A	SINI1 ('blue')
		Comparative	SC/AC	MEN6WI1 ('lesser')
		Superlative	SC/AS	NOVE1WI1 ('newest')
Participles	Active	Present	SC/LAN	PIWU5I1 ('writing')
		Past	SC/LAP	PISAVWI1 ('who wrote')
	Passive	Present	SC/LPN	CITAEMY1 ('being read')
		Past	SC/LPP	POLUCENNY1 ('received')
Pronominal adjectives (pronouns acting as adjectives)			SC/P	NAW ('our') INO1 ('other')

RG2 rules

The existing rules require that adjectives agree in case (CP) and number-gender (NG) (5). As already noted in Section 2.1.3, the resultant adjectival phrase (AP) requires the existance of another rule (6). In both (5) and (6), attributes not mentioned in the rules should be copied from the C_2 (ETC/2).

(5) A CP/X NG/Y
+ A CP/X NG/Y
= AP CP/X NG/Y ETC/2

(6) A CP/X NG/Y
+ AP CP/X NG/Y
= AP CP/X NG/Y ETC/2

Proposed rules

In line with suggestions made in Section 2. 1. 3, the above rules can be replaced by (7). The tag AAX1, whose meaning is "the first exclusion tag in adjective-adjective rules", is used to enforce right-branching bracketing.

(7) A CP/X NG/Y AAX1/*
+ A CP/X NG/Y
= A CP/X NG/Y AAX1/PLUS ETC/2

Note: Additional restrictions on the type of adjectives admitted to this rule would have to be included at the time of actual implementation.

2. Cardinal Numeral Strings

The subclasses of numerals in RG2, derived according to the nature of their ties with nouns, are shown in Table II-2.

RG2 rules

RG2 rules, which only partially reflect the agreement requirements stated in Table II-2, are presented with a minimum of comment, since their limitations will become apparent in the course of the discussion that follows.

(8) C CP/X
+ C CP/X SC/Y-M
= CP CP/X SC/Y ETC/2

Comments: If two cardinal numerals (C) agree in case (CP) and C_2 is of a subclass (SC) other than M (million and higher), they can be combined to form a cardinal numeral phrase (CP) which is in the same case and which acquires the subclass of the rightmost (C_2) numeral as well as all of its other attributes not mentioned in the subrule (ETC/2).

(9) C CP/X-G, D, I, L GC/Y
+ C SC/M CP/Y
= CP SC/M CP/X GC/G ETC/2

Comments: Any cardinal numeral whose case is other than genitive, dative, instrumental or locative (hence, accusative or nominative) can govern an SC/M numeral in the corresponding case (GC/Y and CP/Y on C_1 and C_2, respectively). The resultant CP is an SC/M numeral, acquires the case of C_1, and can govern the genitive (GC/G).

Table II-2: Subclassification of Cardinal Numerals in RG2

Subclass of the numeral	Definition of mnemonics	Comments about numeral-noun agreement and government[21]
SC/S1[22]	Numeral 1 and all numerals ending in -1, except 11.[23]	Numerals in this category agree in case, number and gender with the noun to which they refer. E.g., SOROK ODIN STOL ('forty-one tables').
SC/S2	Numeral 2 and all numerals ending in -2, except 12.	In the nominative (and in the accusative if identical with the nominative) the numeral governs the noun in genitive singular and agrees in gender with the noun: DV4 STOLA ('two tables'); DVE DAMY ('two ladies').
SC/S3	Numeral 3 or 4 and all numerals ending in -3, -4, except 13 and 14.	Preceding comment applies. However, there is no agreement in gender: STO CETYRE STOLA ('104 tables'); STO CETYRE DAMY ('104 ladies').
SC/S5	Numerals 5-19, even tens (20, 30, ...), and all those ending in -5--9.	Preceding comment applies. However, the governed noun must be in the genitive plural. E.g., P4T6 STOLOV ('five tables').
SC/H	Even hundreds or numerals ending in even hundreds.	Preceding comment applies. However, the "nounness" of STO ('hundred') comes to the fore and in oblique cases, typically instrumental, a government variant is possible (noun in the genitive plural) DVUM4STAMI RUBL4MI/RUBLE1 ('200 rubles').
SC/T	Even thousands or numerals in even thousands.	Since TYS4CA ('thousand') can be both a numeral and a numeral noun, the tendency noted immediately above is fully within the literary norm[24]: TYS4CE STOLOV ('thousand tables').
SC/M	Even millions, billions, etc.[25]	These numerals behave like nouns and govern nouns in the genitive plural[24]: MILLION STOLOV ('million tables').

```
(10)  C CP/X-N, A
      + C SC/M CP/X
      = CP SC/M CP/X ETC/2
```

Comment: This subrule states that any cardinal numeral whose case is other than nominative or accusative can be combined with an immediately following SC/M numeral agreeing with it in case.

RG2 contains a CP + C = CP type rule with three subrules having the same restrictions as those illustrated in (8)-(10) above. The left-branching bracketing thus enforced $(((C_n\ C_{n+1})C_m))$ could be easily obtained in another way if the following additional restrictions were introduced into (8)-(10):

```
(11)  C ......
      + C ...... CCX1/*
      = C ...... CCX1/PLUS
```

Comments: CCX1 is a locally important tag condition which can be called "first exclusion tag in numeral-phrase rules". This is the same approach employed in (7), where right-branching bracketing is enforced.

Moreover, if the rules modified as shown in (11) were ordered as in (12) and (13), it would be possible to eliminate subrule (10). Note that some minor adjustments have been made; deletions are shown in brackets and additions are underlined to facilitate comparison with (8) and (9).

```
(12)  C CP/X
      + C CP/X [SC/Y-M] CCX1/*
      = C CP/X [SC/Y] CCX1/PLUS ETC/2 (S/QUIT F/(13))

(13)  C CP/X-G, D, I, L GC/Y
      + C SC/M, T SC/Z CP/Y CCX1/*
      = C SC/Z CP/X GC/G CCX1/PLUS ETC/2 (S/QUIT F/QUIT)
```

Comments: The S and F "tags" are actually instructions which indicate which subrule the recognition routine should process next in cases of "success" or "failure" of the given subrule. The "value" QUIT indicates that the search pass through the subrule packet should be terminated. The "value"(13)of F in (12) is meant to indicate that subrule (13) in this section is to be processed next if subrule (12) fails. In actual practice, the value of each success or failure transfer (S or F), as well as the subrule identifier to which it refers, would have to be a symbol of the tag metalanguage and hence would have to begin with an alphabetic character. The latter restriction will not be observed in this presentation. The repetition of SC in the C_2 line in (13), first with a list of acceptable constants and then with a variable, is an alternate method of enforcing the type of restriction illustrated by the value exclusions of CP in the C_1 line of the same subrule. Additions and deletions in (12) and (13) involving items other than CCX1, S, and

F will remedy certain defects of (8) and (9). In order to correct other limitations of the existing rules, a set of rules similar to the following might be proposed.

Proposed rules

The distinctions expressed in Table II-2 reflect only the grammatical properties of numerals. In order to insure that only lexically permissible sequences of numerals are combined, as well as for some other purposes,[26] it is necessary to introduce an attribute which will be called RANK. The values of this attribute and the groups of numerals to which they are assigned are shown in Table II-3.

Table II-3: Value of the Attribute RANK

Group A: numerals		Group B: numeral nouns	
Value of RANK	Numerals	Value of RANK	Numerals
ONES	1-9	THOUSD	1,000
TEENS	10-19	MILION	1,000,000
TENS	20, 30,..., 90	BILION	1,000,000,000
HUNDRD	100, 200, ..., 900		

The normally permissible order of concatenation of numerals would require the value of RANK on successive constituents to be restricted as shown in Table II-4.

Table II-4: Restrictions on Concatenation of Numerals

The following values of RANK cannot appear on C_1 when the value of the same attribute on C_2 is as shown in the next column	Value of RANK on C_2
ONES, TEENS	ONES
ONES, TEENS, TENS	TEENS
ONES, TEENS, TENS	TENS
ONES, TEENS, TENS, HUNDRD	HUNDRD
THOUSD	THOUSD
THOUSD, MILION	MILION
THOUSD, MILION, BILION	BILION

When expressed in CSA rule format, the above restrictions give rise to the packet of ordered subrules presented immediately below (14.1)-(14.6). All of the subrules involve partial tests and therefore result in a dummy constituent (DUMMY) which is introduced solely for the purpose of satisfying subrule format requirements. The values of S and F instructions refer to numbers of subrules in this text. In cases of success (S), the search is continued in the next packet of subrules, starting with (15. 1).

```
(14. 1)   C RANK/X-ONES, TEENS
          + C RANK/ONES
          = DUMMY (S/(15. 1) F/(14. 2))

(14. 2)   C RANK/X-ONES, TEENS, TENS
          + C RANK/TEENS, TENS
          = DUMMY (S/(15. 1) F/(14. 3))

(14. 3)   C RANK/ONES, TEENS, TENS, HUNDRD
          + C RANK/HUNDRD
          = DUMMY (S/(15. 1) F/(14. 4))

(14. 4)   C RANK /X-THOUSD
          + C RANK/THOUSD
          = DUMMY (S/(15. 1) F/(14. 5))

(14. 5)   C RANK/X-THOUSD, MILION
          + C RANK/MILION
          = DUMMY (S/(15. 1) F/(14. 6)

(14. 6)   C RANK/X-THOUSD, MILION, BILION
          + C RANK/BILION
          = DUMMY (S/(15. 1) F/QUIT)
```

Subrules (14. 1)-(14. 5) deal only with the problem of concatenation. In order to describe the grammatical links which must be considered in addition, it will be convenient first to introduce some additional notation characterizing classes of numeral strings. Let cardinal numerals and combinations of numerals and numeral nouns formed according to the restrictions stated in (14. 1)-(14. 6) be denoted by two-character symbols beginning with "M", with MS representing a single numeral (Group A in Table II-3) and MN a numeral noun (Group B in Table II-3). Further, let M1-M4 represent combinations of numerals and numeral nouns differentiated according to (a) presence or absence of numeral nouns and (b) the relative position of the numeral noun, factors which determine the nature of the grammatical tests that must be carried out when checking the validity of each potential combination. The information presented in Table II-5 is expressed in an ordered set of subrules (15. 1)-(15. 5) below. For the sake of illustration, the "M-symbol" appears as the value of the attribute MTYPE - "M-type". (Although the use of this tag could be avoided through employment of an elaborate system of exclusion tags, the resultant rules would be much more difficult to follow and to describe.) The five subrules (15. 1)-(15. 5) constitute five "steps" that determine which of four "tests" is necessary to complete the recognition of a numeral string.

Table II-5: Permissible Combinations of M-Constituents

line	Hypothetical rule	Test	Russian example	Numeric equivalent
a.	MS + MS = M1	1	SOROK TRI	43
b.	MS + M1 = M1	1	STO SOROK TRI	143
c.	MS + MN = M2	2	TRI TYS4CI	3,000
d.	M1 + MN = M2	2	SOROK TRI TYS4CI	43,000
e.	M2 + MS = M3	4	TRI TYS4CI DVA	3,002
f.	M2 + M1 = M3	4	TRI TYS4CI SOROK TRI	3,043
g.	M2 + M2 = M3	4	DVA MILLIONA SOROK TYS4C	2,040,000
h.	M2 + M3 = M3	4	DVA MILLIARDA MILLION SOROK TRI TYS4CI	2,001,043,000
i.	M2 + MN = M3	4	DVA MILLIONA TYS4CA	2,001,000
j.	MN + MS = M3	3	TYS4CA DVA	1,002
k.	MN + M1 = M3	3	TYS4CA SOROK TRI	1,043
l.	MN + M2 = M3	4	MILLION SOROK TRI TYS4CI	1,043,000
m.	MN + M3 = M3	4	MILLIARD ODIN MILLION TYS4CA SOROK TRI	1,001,001,043
n.	MN + MN = M3	4	MILLION TYS4CA	1,001,000
o.	MN + MS = M4	3	TYS4CI DVE	about 2,000
p.	MN + M1 = M4	3	TYS4CI SOROK TRI	about 43,000

Step 1. If the C_1 is an MS and C_2 is MS or M1, go to Test 1 - subrule (16. 1); otherwise (15. 2).

```
(15. 1)   C MTYPE/MS
          + C MTYPE/MS, M1
          = DUMMY (S/(16. 1) F/(15. 2))
```

Step 2. If the C_1 is an MS or M1 and C_2 is an MN, go to Test 2 - subrule (16. 2); otherwise (15. 3).

```
(15. 2)   C MTYPE/MS, M1
          + C MTYPE/MN
          = DUMMY (S/(16. 2) F/(15. 3))
```

Step 3. If the C_1 is an MN and C_2 is an MS or M1, go to Test 3 - subrule (16. 7); otherwise (15. 4).

```
(15. 3)   C MTYPE/MN
          + C MTYPE/MS, M1
          = DUMMY (S/(16. 7) F/(15. 4))
```

Step 4. If the C_1 is an M2 and C_2 is any M, except M4, go to Test 4 - subrule (16. 12); otherwise (15. 5).

```
(15. 4)   C MTYPE/M2
          + C MTYPE/X-M4
          = DUMMY (S/(16. 12) F/(15. 5))
```

Step 5. If C_1 is an MN and C_2 is any M, except M4, MS, or M1, go to Test 4 - subrule (16. 12); otherwise QUIT.

```
(15. 5)   C MTYPE/MN
          + C MTYPE/X-M4, MS, M1
          = DUMMY (S/(16. 12) F/QUIT)
```

Test 1. Here only agreement in case is required. The value of RANK on C_1 is copied onto the corresponding attribute of C_3 since this information is necessary for the tests in (14. 1)-(14. 6), which will be reapplied when attempting to form still longer chains. The value of RANK on C_2 is copied onto RANKRT - "rank on the right" in order to preserve information necessary for other rules which may subsequently apply.[26] CCX3 is a locally important tag which, like other CCX-prefixed tags, is used to insure a given order of concatenation.

```
(16. 1)   C CP/X RANK/Y CCX3/*
          + C CP/X RANK/Z
          = C MTYPE/M1 CP/X RANK/Y RANKRT/Z
            CCX3/PLUS ETC/2 (S/QUIT F/QUIT)
```

Test 2. In (16.2), provisions are made for S1 numerals which, as shown in Table II-2, must agree in case (CP), number and gender (NG) with the numeral noun. In (16.3), all other numerals in oblique cases need agree only in case. Subrules (16.4)-(16.6) test for situations where the first numeral is in the nominative or accusative case (CP/N, A) and governs the second numeral in the genitive case (CP/G) in accordance with the conditions stated in Table II-2. Hence, in (16.4), which applies to S2 numerals, agreement in gender is required; in (16.5), agreement in gender is not required, but the noun must be singular; in (16. 6), the tests of (16.5) are repeated with the exception of the number (NG) of C_2 which must be plural. Since (16.6) is the last rule of the particular subpacket, further search is terminated in instances of either success (S/QUIT) or failure (F/QUIT).

```
(16. 2)  C SC/S1 CP/X NG/Y RANK/XA
         + C CP/X NG/Y RANK/XB
         = C MTYPE/M2 CP/X NG/Y RANK/XA RANKRT/XB
           ETC/2 (S/QUIT F/(16. 3))

(16. 3)  C SC/XC-S1 CP/X-N, A CP/Z RANK/XA
         + C CP/Y-N, A CP/Z RANK/XB
         = C MTYPE/M2 CP/Z RANK/XA RANKRT/XB ETC/2
           (S/QUIT F/(16. 4))

(16. 4)  C SC/S2 CP/N, A CP/X NG/Y RANK/XA
         + C CP/G NG/Y-P RANK/XB
         = C MTYPE/M2 CP/X NG/Y GC/G RANK/XA
           RANKRT/XB ETC/2 (S/QUIT F/(16. 5))

(16. 5)  C SC/S3 CP/N, A CP/S NG/Y RANK/XA
         + C CP/G NG/Z-P RANK/XB
         = C MTYPE/M2 CP/X NG/Y GC/G RANK/XA
           RANKRT/XB ETC/2 (S/QUIT F/(16. 6))

(16. 6)  C SC/S5, H CP/N, A CP/X NG/Y RANK/XA
         + C CP/G NG/P RANK/XB
         = C MTYPE/M2 CP/X NG/Y GC/G RANK/XA
           RANKRT/XB ETC/2 (S/QUIT F/QUIT)
```

Test 3. The tests in this packet of subrules are intended for the recognition of approximate quantities shown in lines "o" and "p" of Table II-5. The restrictions on the use of inversion of normal numeral-noun sequence to express approximation are not clear and the conditions imposed here are accordingly based purely on Sprachgefühl.

In oblique cases it is necessary to permit both an M3 and an M4 to be generated. Subrules (16. 7) and (16. 8) require case agreement only.

```
(16.7) C CP/Z RANK/XA
       + C CP/Z RANK/XB
       = C MTYPE/M3 CP/Z RANK/XA RANKRT/XB ETC/2
         (S/(16.8) F/(16.9))

(16.8) C CP/X-N,A CP/Z RANK/XA
       + C CP/Y-N,A CP/Z RANK/XB
       = C MTYPE/M4 CP/Z RANK/XA RANKRT/XB ETC/2
         (S/QUIT F/QUIT)
```

The next three subrules reverse the order of C_1 and C_2 of the corresponding subrules (16.4)-(16.6). The value of MTYPE is set to M4.

```
(16.9)   C CP/G NG/Y-P RANK/XA
         + C SC/S2 CP/N,A CP/X NG/Y RANK/XB
         = C MTYPE/M4 CP/X NG/Y GC/G RANK/XA
           RANKRT/XB ETC/1 (S/QUIT F/(16.10))

(16.10)  C CP/G NG/Z-P RANK/XA
         + C SC/S3 CP/N,A CP/X NG/Y RANK/XB
         = C MTYPE/M4 CP/X NG/Y GC/G RANK/XA
           RANKRT/XB ETC/1 (S/QUIT F/(16.11))

(16.11)  C CP/G NG/P RANK/XA
         + C SC/S5,H CP/N,A CP/X NG/Y RANK/XB
         = C MTYPE/M4 CP/X NG/Y GC/G RANK/XA
           RANKRT/XB ETC/1 (S/QUIT F/QUIT)
```

Test 4. This test repeats the requirements of (16.1) in Test 1; however, the value of MTYPE is set to M3 rather than M1.

```
(16.12)  C CP/X RANK/Y
         + C CP/X RANK/Z
         = C MTYPE/M3 CP/X RANK/Y RANKRT/Z ETC/2
           (S/QUIT F/QUIT)
```

Note: The CCX3 is unimportant in (16.12) and is therefore not used.

Comparison of RG2 and proposed rules

Although, as was shown above, the six subrules for recognition of cardinal numerals contained in RG2 could have been reduced to two, (12) and (13), the more comprehensive treatment of the problem proposed here would require some twenty-three subrules. The detailed treatment of numeral strings in the illustrations is itself incomplete, since it neglects some of the problems which can be caused by the non-standard usage of SC/H numerals (hundreds) and does not fully consider the functions of SC/T numerals (thousands). However, these gaps appear to be relatively insignificant.

3. Noun-Noun Predications

Two nouns in the nominative case which, in addition, usually agree in number, gender, and animateness can combine to form a predication or an apposition. By way of general comment, two nouns are seldom encountered either without appropriate punctuation (a dash) or additional words such as the negative particle NE when one of them acts as a subject and the other as the corresponding predicate.[27] Thus, $FIZIKA--NAUKA ('Physics is a science') or $ALXIMI4 NE NAUKA ('Alchemy is not a science'). It is only with shorter sentences and then in instances approaching colloquial usage that sentences like $MO1 BRAT UCITEL6 ('My brother is a teacher') are possible. Primarily for this reason, RG2 did not contain a specific N + N subrule producing a predication.

4. Close Appositions

The Russian term for appositives (prilozhenie) is a calque from the Latin appositio. However, related etymologies notwithstanding, the respective usage of the two terms differs considerably in grammars of English and of Russian.[28] The use of the terms appositive and apposition in the present context is limited to constructions described in the present subsection.[29] In order to develop grammar rules for the recognition of close appositions, i.e., those whose components are not separated by punctuation, a detailed subclassification of nouns is required. Only a broad outline of such a subclassification is presented at this time.

The subclassification proposed in Figure II-1 is based on requirements of Russian grammar; stylistic considerations have influenced only the relative emphasis on certain construction types. Examples illustrating the subclasses shown in Figure II-1 and one possible way in which the relevant information can be expressed in tag notation appear in Table II-6. The tags introduced in Table II-6 are defined in Table II-7.

Subgroups of close appositions

The characteristics of the twelve main subgroups within close appositions, summarized in Table II-8, are briefly commented on below. In each instance, a sketch of the linguistic background is followed by proposed recognition rules.

A1 Subgroup: Two personal names

The only instance of a true personal name appositive construction[42] is not discussed here. However, for functional reasons, it is convenient to create quasi-appositions consisting of personal names. Russian and foreign names are considered separately because of differences in patterns of

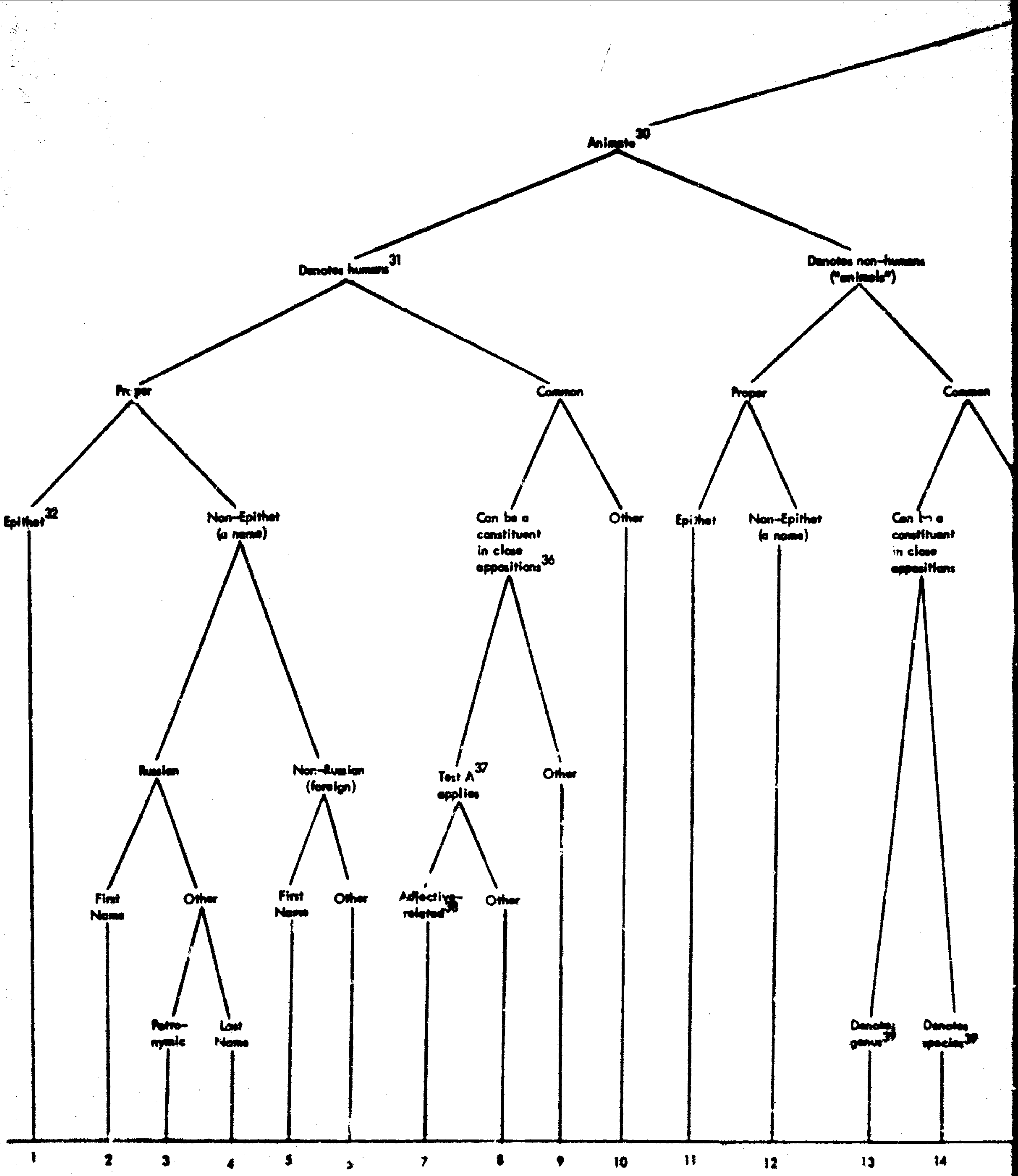
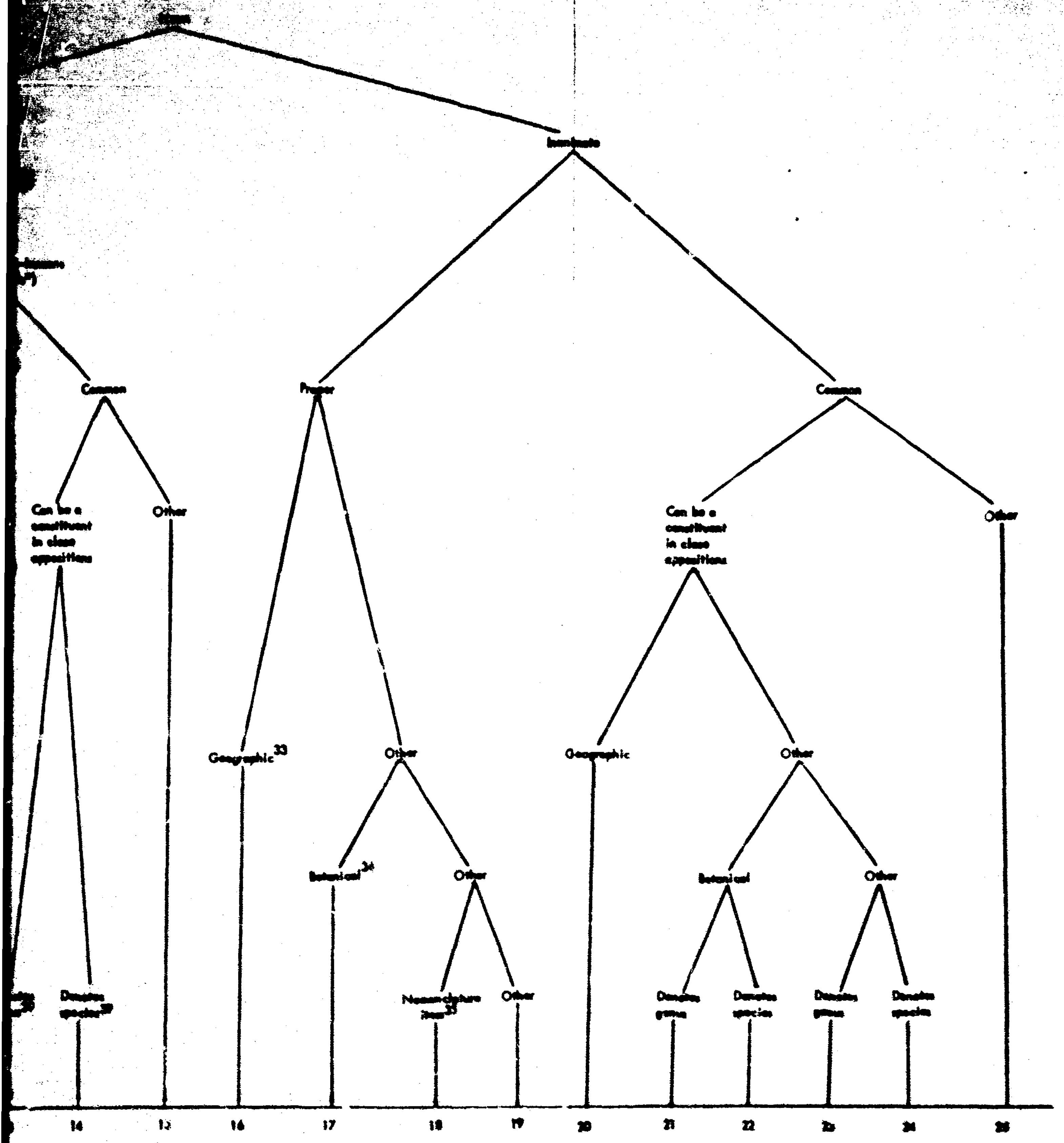


Figure II-1: SUBCLASSES



t SUBCLASSES OF RUSSIAN NOUNS

2

Table II-6: Examples of Subclasses Defined in Figure II-1

Noun sub-class	Russian words	English translation	Tags describing the noun subclass (defined in Table II-7)
1	$VSEVOLOD $BOL6WOE $GNEZDO	Vsevolod the Large Nest	ANIMTE/HUMAN NCLASS/EPITH
2	$IVAN, $PETR, $OL6GA, $ALEKSANDR	Ivan, Peter, Olga, Alexander	ANIMTE/HUMAN NAME/FIRST NCLASS/PROPER RUSIAN/PLUS
3	$IVANOVIC, $PETROVNA, $ALEKSANDROVIC	Ivanovich, Petrovna, Alexandrovich	ANIMTE/HUMAN NAME/PATRO NCLASS/PROPER RUSIAN/PLUS
4	$PETROV, $JAROV, $PLHWKIN	Petrov, Zharov, Pliushkin	ANIMTE/HUMAN NAME/LAST NCLASS/PROPER RUSIAN/PLUS
5	$FRANKLIN, $3LEANORA, $ROBERT	Franklin, Eleanor, Robert	ANIMTE/HUMAN NAME/FIRST NCLASS/PROPER FOREGN/PLUS
6	$RUZVEL6T, $RICARDSON, $VIARDO[40]	Roosevelt, Richardson, Viardot	ANIMTE/HUMAN NAME/LAST NCLASS/PROPER FOREGN/PLUS
7	STARIK, BOGAC, KRASAVIQA	old man, rich person, a beauty	ANIMTE/HUMAN NCLASS/COMMON GROUPA/PLUS TESTA/ADJREL
8	TOVARI5, GRAJDANIN, GOSPODIN	comrade, citizen, mister	ANIMTE/HUMAN NCLASS/COMMON GROUPA/PLUS TESTA/NONADJ
9	INJENER, LETCIK, STUDENT	engineer, pilot, student	ANIMTE/HUMAN NCLASS/COMMON GROUPA/PLUS
10	CELOVEK, JEN5INA, MUJCINA[41]	man, woman, man	ANIMTE/HUMAN NCLASS/COMMON GROUPA/MINUS
11	JEREBEQ $KOLESO	the stallion Koleso	ANIMTE/ANIMAL NCLASS/EPITH
12	$KAWTANKA, $JUCKA	Kashtanka, Zhuchka	ANIMTE/ANIMAL NCLASS/PROPER

Table II-6 (contd.)

Noun sub-class	Russian words	English translation	Tags describing the noun subclass (defined in Table II-7)
13	RYBA, PTIQA, JIVOTNOE	fish, bird, animal	ANIMTE/ANIMAL GROUPA/PLUS GENUS/ANIMAL NCLASS/COMMON
14	AKULA, LASTOCKA, VOLK	shark, swallow, wolf	ANIMTE/ANIMAL GROUPA/PLUS SPCIES/ANIMAL NCLASS/COMMON
15	No examples found		
16	$MOSKVA, $VOLGA, $ITALI4	Moscow, Volga, Italy	INANIM/GEOGR NCLASS/PROPER
17	$DIANA $CERNENKO	Diana Chernenko (name of an apple tree)	INANIM/BOTAN NCLASS/PROPER
18	MI-18	MI-18 (a helicopter)	INANIM/NOMEN NCLASS/NOMEN
19	An empty subclass		
20	GOROD, REKA, STRANA	city, river, country	INANIM/GEOGR NCLASS/COMMON GROUPA/PLUS
21	DEREVO, RASTENIE	tree, plant	INANIM/BOTAN NCLASS/COMMON GROUPA/PLUS GENUS/BOTAN
22	KIPARIS, 4BLON4	cypress, apple tree	INANIM/BOTAN NCLASS/COMMON GROUPA/PLUS SPCIES/BOTAN
23	GAZ	gas	INANIM/OTHER NCLASS/COMMON GROUPA/PLUS GENUS/OTHER
24	BUTAN	butane	INANIM/OTHER NCLASS/COMMON GROUPA/PLUS SPCIES/OTHER
25	VSELENNA4, NASELENIE	the universe, population	INANIM/OTHER NCLASS/COMMON GROUPA/MINUS

Table II-7: Attributes and Values of Tags Used in Table II-6

Attribute		Possible values
Name	Meaning	(meaning of each value follows in parentheses)
ANIMTE	Animate	ANIMAL (animal), HUMAN (human)
FOREGN	Foreign	PLUS (plus)
GENUS	Genus	ANIMAL (animal). BOTAN (botanical), OTHER (other)
GROUPA	Close Apposition	PLUS (plus), MINUS (minus)
INANIM	Inanimate	BOTAN (botanical), GEOGR (geographical), NOMEN (nomenclature item), OTHER (other)
NAME	Name	FIRST (first name), LAST (last name), PATRO (patronymic)
NCLASS	Noun class	COMMON (common noun), EPITH (epithet), NOMEN (nomenclature item), PROPER (proper name)
RUSIAN	Russian	PLUS (plus)
SPCIES	Species	ANIMAL (animal), BOTAN (botanical), OTHER (other)
TESTA	Test A	ADJREL (adjective-related), NONADJ (not adjective-related)

Table II·8: Subgroups of Close (Group A) Appositions

Value of the Attribute NCLASS		For both nouns				
of C_1	of C_2	following attributes	have values	sub-group	other tags	Examples or Comments
PROPER	PROPER	ANIMTE	HUMAN	A1	(1)	Russian personal names
PROPER	PROPER	ANIMTE	HUMAN	A1	(1)	Foreign personal names
COMMON	PROPER	ANIMTE	HUMAN	A2	(2)	INJENER $PETROV ('engineer Petrov' (masc.))
COMMON	PROPER	ANIMTE	HUMAN	A2	(2)	UCITEL6NIGA $RICARDS ('teacher (fem.) Richards')
COMMON	PROPER	ANIMTE	HUMAN	A2	(2)	INJENER $L4POVA ('engineer Liapova' (fem.))
COMMON	PROPER	ANIMTE	ANIMAL	A3	(2)	SOBAKA $JUCKA ('dog Zhuchka')
COMMON	PROPER	INANIM	GEOGR	A4	(2)	GOROD $OMSK ('city of Omsk')
COMMON	PROPER	INANIM	GEOGR	A4	(2)	OZERO $BAIKAL ('lake Baikal')
COMMON	PROPER	INANIM	BOTAN	A5	(2)	4BLON4 $DIANA $CERNENKO ('Diana Chernenko apple tree')
COMMON	COMMON	ANIMTE	HUMAN	A6	(3)	STARIK PIRAT ('old ('man') pirate')
COMMON	COMMON	ANIMTE	HUMAN	A6	(4)	TOVARI5 MINISTR ('comrade minister')
COMMON	COMMON	ANIMTE	ANIMAL	A7	(5)	RYBA AKULA ('the fish shark')
COMMON	COMMON	INANIM	BOTAN	A8	(5)	DEREVO KIPARIS ('cypress tree')
COMMON	COMMON	INANIM	OTHER	A9	(5)	GAZ BUTAN ('the gas butane')
PROPER	EPITH	ANIMTE	HUMAN	A10		$VSEVOLOD $BOL6WOE $GNEZDO ('Vsevolod the Large Nest')
COMMON	EPITH	ANIMTE	ANIMAL	A11		JEREBEQ $KOLESO ('the stallion Koleso')
COMMON	NOMEN	INANIM	any	A12		VERTOLET MI-18 ('the MI-18 helicopter') TRIOD PT-1 ('PT-1 triode')

Table II-8: Notes to column "other tags"

(1) See discussion of A1 subgroup.

(2) The first noun must have the attribute GROUPA.

(3) The first noun must have the tag TESTA/ADJREL.

(4) The first noun must have the tag TESTA/NONADJ.

(5) The first noun must have the attribute GENUS and the second noun the tag SPCIES, both with the value of the ANIMTE or INANIM attribute, as the case may be

- -

agreement.

Russian personal names

First names (NAME/FIRST), patronymics (NAME/PATRO), and last names (NAME/LAST) have been identified in Figure II-1 and Table II-6. In addition, a constituent consisting of a first name and a patronymic (NAME/FPATRO) must also be considered. If $IVAN ('Ivan'), $IVANOVIC ('Ivanovich'), and $PETROV ('Petrov') are taken as representative examples, the following five constructions are possible.

a. First name and patronymic -- $IVAN $IVANOVIC

b. First and last name -- $IVAN $PETROV

c. First name, patronymic, last name -- $IVAN $IVANOVIC $PETROV

d. Last name, first name -- $PETROV $IVAN

e. Last name, first name, patronymic -- $PETROV $IVAN $IVANOVIC

All items in a-e above must agree in case, number, and gender.

Foreign personal names

Only constructions g and h below will be discussed here.

g. Several first names agreeing in case followed by a last name in the same case. For instance, $FRANKLIN $ROBERT $RICARDSON ('Franklin Robert Richardson').

h. One or more first names agreeing in case followed by a foreign feminine last name (which is identical to the nominative masculine form). For example, $3LEANORA $RUZVEL6T ('Eleanor Roosevelt').

i. The version of both constructions where the last name occurs first is possible in principle, but is not considered here despite the fact that some constructions could be recognized by subrule (24) below.

Proposed rules for A1 subgroup

Step 1. It is necessary to establish that the two nouns (N) one is to deal with in this packet are both proper names (NCLASS/PROPER) denoting human beings (ANIMTE/HUMAN).

(17) N NCLASS/PROPER ANIMTE/HUMAN
+ N NCLASS/PROPER ANIMTE/HUMAN
= DUMMY (S/(18) F/(24))

Step 2. In the next two subrules, agreement tests are performed.

(18) N CP/X NG/Y
+ N CP/X NG/Y
= DUMMY (S/(20) F/(19))

Comment: If the two nouns agree in case (CP) and number-gender (NG), go to subrule (20); otherwise (19).

(19) N CP/X NG/F NAME/FIRST
+ N CP/N NG/F NAME/LAST FOREGN/PLUS NNX1/*
= N CP/X NG/F NAME/LAST FOREGN/PLUS
NNX1/FIRST ETC/2 (S/QUIT F/QUIT)

Comment: This subrule is designed to handle (g) and (h) above. Any feminine (NG/F) first name (NAME/FIRST) in any case (CP/X) can combine with a foregin (FOREGN/PLUS) feminine (NG/F) last name (NAME/LAST) which is in the nominative (CP/N) and does not have the tag NNX1/FIRST. The C_3 is a foreign feminine last name in the same case as the C_1; all other tags not mentioned in the subrule are copied from the C_2 (ETC/2). NNX1 is a locally important tag (the first exclusion tag in noun-noun rules) which enforces left-branching bracketing.

Step 3. Next, proper sequence of first names, patronymics, last names, and their combinations is enforced by subrules (20)-(23).

(20) N NAME/FIRST NNX1/*
+ N NAME/FIRST
= N NAME/FIRST NNX1/FIRST ETC/2 (S/QUIT F/(21))

Comment: Subrule normally applies only to foreign names. Any number of first names (g) can be combined through repeated applications of (20), resulting in a first name which has the tag NNX1/FIRST. Since there are no restrictions regarding NNX1 on C_1 in (19), that subrule can then be used to recognize varieties of (g) and (h) above.

(21) N NAME/FIRST NNX1/*
+ N NAME/PATRO
= N NAME/FPATRO ETC/2 (S/QUIT F/(22))

Comment: Since a patronymic (NAME/PATRO) can only combine with a single first name, output of (20) is excluded (NNX1/* on C_1). This subrule is designed to handle (a) above.

(22) N NAME/FIRST, FPATRO NAME/X
+ N NAME/LAST NNX1/*
= N NAME/LAST NNX1/X ETC/2 (S/QUIT F/(23))

Comment: This subrule is similar to (19) and is intended to recognize (b), (c), and (g). Any first name or (first name)-patronymic construction can combine with a last name not produced by (19), (22), or (23). The tag NAME is repeated on C_1 in order to copy its value onto NNX1 on C_3.

(23) N NAME/LAST NNX1/*
+ N NAME/FIRST, FPATRO NAME/X
= N NAME/LAST NNX1/X ETC/2 (S/QUIT F/QUIT)

Comment: The C_1-C_2 sequence of (22) is reversed here. C_3 differs in the value of the F instruction because this is the last subrule of the packet. Items in (d) and (e) are recognized by this rule, as are items in (i) corresponding to (g), or to instances of (h) where all constituents are in the nominative case.

A2 and A3 Subgroups: Common noun - proper name, both animate

The common noun is usually a "title" describing a human being by reference to profession, social, political, military and other ranks, national or regional origin, personal characteristics or qualities, and the like (lines 7-9 of Table II-6). Any personal name in lines 2-6 of Table II-6 or A1 subgroup quasi-apposition can be used as the second component. Varieties of agreement are affected by the peculiarities of the proper name. Instances where A2-appositions contain several "titles" are not discussed;[43] the same applies to instances where C_1 is itself an apposition (Cf. note 29, example (6)).

Proposed rules for A2 and A3 subgroups

The packet of subrules described below contains the subrules necessary for the recognition of constructions in the A2 and A3 subgroups.

Step 1. Before establishing the necessary agreement requirements in Step 2, it is necessary to ascertain that the pair of nouns meets the broad requirements of the subrule packet (next three subrules).

(24) N NCLASS/COMMON GROUPA/PLUS
+ N NCLASS/PROPER
= DUMMY (S/(25) F/(35))

Comment: If C_1 is a common noun (NCLASS/COMMON) and can be a member of close (Group A) appositions (GROUPA/PLUS) and if C_2 is a proper name (NCLASS/PROPER), go to (25); otherwise, go to (35).

(25) N ANIMTE/HUMAN
+ N ANIMTE/HUMAN
= DUMMY (S/(27) F/(26))

Comment: If both nouns denote human beings (ANIMTE/HUMAN) and hence may belong to subgroup A2, go to subrule (27); otherwise, go to (26).

(26) N ANIMTE/ANIMAL
+ N ANIMTE/X
= DUMMY (S/(30) F/(31))

Comment: If the C_1 is an "animal" (ANIMTE/ANIMAL), C_2 can be either animal or human (ANIMTE/X). If (26) is successful, indicating potential membership in subgroup A3, subrule (30) should be accessed; otherwise go to (31).

Step 2. The next four subrules establish agreement requirements.

(27) N CP/X NG/Y APPOS/*
+ N CP/X NG/Y
= N CP/X NG/Y APPOS/A2 ETC/1 (S/(47) F/(28))

Comment: This subrule imposes the same agreement requirements as does (18). In order to restrict C_1 to a simple common noun, the tag APPOS/* is employed. Note that although an apposition is created, it is necessary to test for a possible second analysis where C_1 governs C_2. For example, DRUG SESTRY $MARII can have two analyses corresponding to (a) 'the friend of sister Mary' and

(b) 'the friend of Mary's sister'. Hence, (S/(47), F/(28)). (Although such possibilities of government may also exist for other subgroups of appositions, they are not explicitly indicated in the transfer sections of subrules for subgroups A4-A12.)

(28) N CP/X NG/M APPOS/*
+ N CP/X NG/F RUSIAN/PLUS
= N CP/X NG/M, F APPOS/A2 ETC/1 (S/(47) F/(29))

Comment: This subrule differs from the preceding one in allowing C_1 to be masculine (NG/M) and C_2 to be feminine (NG/F).[44] However, C_2 (the proper name in this instance) must be a Russian personal name (RUSIAN/PLUS). (Cf. the example INJENER $L4POVA.)

(29) N CP/X NG/Y-P APPOS/*
+ N CP/N NG/F FOREGN/PLUS
= N CP/X NG/Y, F APPOS/A2 ETC/1 (S/QUIT F/(30))

Comment: This subrule is similar to (19) in the type of agreement requirements. Subject to (24), it allows a foreign feminine name in the nominative case to combine with a noun in any case (CP/X) and of any gender agreeing with it in number (NG/Y-P). C_3 acquires the case, the number-gender, and all other attributes of C_1. However, in addition to all other possible values of NG on C_3 it also will have feminine (F).

(30) N CP/X NG/Y-P ANIMTE/ANIMAL APPOS/*
+ N CP/X NG/Z-P
= N CP/X NG/Y ANIMTE/ANIMAL APPOS/A3 ETC/1 (S/(47) F/(47))

Comment: This subrule is restricted in its application to names given to animals, which may (on rare occasions) disagree in gender with the common noun.

A4 Subgroup: Geographical designations

The two options of agreement between nouns in this construction[45] are generally inadequately distinguished according to differences in usage in Russian grammar and are here considered interchangeable. The features shown for constituents in this subgroup of appositions are insufficient to avoid the problems mentioned in note 39: a sentence like ZA GORO1 $VOLGA STANOVITS4 XOLODNEE can have two analyses corresponding to (a) 'Beyond the mountain, the Volga becomes cooler' and (b) 'Beyond Mt. Volga it becomes cooler'.[46]

Proposed rules

The packet of subrules required for recognition of A4 appositions will be described together with those for the A5 subgroup.

A5 Subgroup: Botanical names

Capitalization of names of botanical species is restricted to specialized texts,[47] with the constructions appearing in the A8 subgroup more commonly use ' instead. C_2 in A5 appositions is always in the nominative.

Proposed rules for the A4 and A5 subgroups

The first subrule of the present packet is accessed following failure of (26).

```
(31)  N INANIM/BOTAN, GEOGR INANIM/X
      + N INANIM/BOTAN, GEOGR INANIM/X
      = DUMMY (S/(32) F/QUIT)
```

Comment: Both nouns must be either botanical or geographic as shown by the repetition of the INANIM - "inanimate" tag.

```
(32)  N INANIM/BOTAN APPOS/*
      + N
      = DUMMY (S/(34) F/(33))
```

Comment: Establish that both nouns are "botanical".

```
(33)  N CP/X NG/Y-P
      + N CP/X NG/Z-P
      = N CP/X NG/Y APPOS/A4 ETC/1 (S/QUIT F/(34))
```

Comment: Handles pairs of geographic nouns. Agreement in case is required. Number must not be plural.[48]

```
(34)  N CP/X
      + N CP/N
      = N CP/X APPOS/A5 ETC/2 (S/QUIT F/QUIT)
```

Comment: The agreement restrictions are similar to those in (29).

A6 Subgroup: Two common nouns (human beings)

The recognition of the two types of constructions in this subgroup would require the following subrules.

```
(35)  N NCLASS/COMMON GROUPA/PLUS
      + N NCLASS/COMMON GROUPA/PLUS
      = DUMMY (S/(36) F/(45))
```

Comment: If the two nouns are common nouns which can be members of Group A appositions (GROUPA/PLUS), go to (36); otherwise, go to (45).

```
(36)  N ANIMTE/HUMAN
      + N ANIMTE/HUMAN
      = DUMMY (S/(37) F/(42))
```

Comment: If both nouns are animate and human, go to (37); otherwise go to (42).

```
(37)  N TESTA/ADJREL
      + N
      = DUMMY (S/(39) F/(38))
```

Comment: If C_1 is of the type STARIK (line 7, Table II-6), go to (39); otherwise, go to (38).

```
(38)  N TESTA/NONADJ
      + N
      = DUMMY (S/(40) F/QUIT)
```

Comment: If C_1 is of the type TOVARI5 (line 8, Table II-6), go to (40); otherwise, QUIT.

```
(39)  N CP/X NG/Y
      + N CP/X NG/Y
      = N CP/X NG/Y APPOS/A6 ETC/2 (S/QUIT F/QUIT)
```

Comment: This subrule requires complete agreement in case (CP), number, and gender (NG). It could have been combined with (37) into a single subrule.

```
(40)  N CP/X NG/Y
      + N CP/X NG/Y
      = N CP/X NG/Y APPOS/A6 ETC/2 (S/QUIT F/(41))
```

Comment: This subrule is identical to (39) except for the fact that in case of failure another subrule (41) should be accessed.

```
(41)  N CP/X NG/Y-P
      + N CP/X NG/Z-P
      = N CP/X NG/Y, Z APPOS/A6 ETC/2 (S/QUIT F/QUIT)
```

Comment: This subrule is similar to (40); gender agreement is relaxed in order to allow such constructions as TOVARI5 MEDSESTRA ('comrade nurse').

A7-A9 Subgroups: Two common nouns (other than human beings)

The subrules for the three subgroups can be considered collectively and are accessed following failure of (36).

(42) N ANIMTE/ANIMAL GENUS/X
+ N ANIMTE/ANIMAL SPCIES/X
= DUMMY (S/(44) F/(43))

Comment: If both nouns are ANIMTE/ANIMAL and C_1 denotes a genus, C_2 a species, go to (44); otherwise, go to (43).

(43) N INANIM/BOTAN, OTHER INANIM/X GENUS/Y
+ N INANIM/BOTAN, OTHER INANIM/X SPCIES/Y
= DUMMY (S/(44) F/QUIT)

Comment: If both nouns agree in the value of the attribute INANIM (inanimate) and C_1 denotes a genus, while C_2 denotes a species, go to (44); otherwise, QUIT.

(44) N CP/X
+ N CP/X
= N CP/X ETC/1 (S/QUIT F/QUIT)

Comment: This subrule could have been incorporated into (42) and (43). Agreement in case is required. In order to require agreement in number, another subrule would have to be added. One subrule would have to have the values of NG set to P, the other to X-P on both the C_1 and C_2 of the respective rules.

A10 and A11 Subgroups: Constructions with an epithet

Since these constructions are rare, only one subrule is given as an illustration.

(45) N NCLASS/COMMON, PROPER NCLASS/X ANIMTE/Y
+ N NCLASS/EPITH ANIMTE/Y
= N NCLASS/X ANIMTE/Y ETC/1 (S/QUIT F/(46))

Comment: If the C_1 is either a proper name or a common noun and C_2 is an epithet agreeing with it in the type of animateness (ANIMTE/Y), a new constituent is produced which is similar to C_1. Otherwise, go to (46).

A12 Subgroup: Constructions with nomenclature items[49]

Although these are very frequently encountered constructions, adequate restrictions are very difficult to work out.[50] The subrule follows.

(46) N NCLASS/COMMON INANIM/X
+ N NCLASS/NOMEN
= N NCLASS/COMMON INANIM/X APPOS/A12 ETC/1
(S/QUIT F/QUIT)

Comment: If an inanimate common noun is left-adjacent to a nomenclature item, the resultant constituent is given the same tags as the common noun.

Comparison of RG2 and proposed rules

In view of the detailed treatment of close appositions presented above, it is necessary to mention only briefly the relevant RG2 rules.

(1) All nouns were divided into three groups:

(a) "titles" -- words of the type INJENER ('engineer') (lines 7-9 of Table II-6) and also those like GOROD (line 20, Table II-6)

(b) "names" -- proper names

(c) all other "regular" nouns.

(2) The only constructions considered were those presented in the A1, A2, and A4 subgroups shown in Table II-8.

(3) Generally speaking, agreement requirements were not as well developed as they are in the proposed rules.

5. Word Groups Formed by a Governing Noun

Government rules are basically simple. As shown in (47), the attributes GC (government) on C_1 and CP (case) on C_2 must be set to the same value.

```
(47)  N CP/Y GC/X
      + N CP/X
      = N CP/Y GC/1-X ETC/1
```

Comment: The value of CP on C_1 is set to another variable in order to copy the value in CP onto C_3; the value of "1-X" of GC on C_3 should be interpreted as follows: "Copy all values of GC on C_1 except X, as defined in the subrule."

Preposition of the governed noun is rare; in expository writing it only occurs in fixed expressions of the type GVARDII LEITENANT ('guards lieutenant'). In RG2, the output of a subrule analogous to (47) resulted in a constituent NP - "noun phrase" which further increased the number of subsequent rules.

6. Nouns With Cardinal Numerals

The (cardinal numeral)-noun constructions parallel those involving numeral nouns (Part A2 of this section), and will consequently not be discussed further here. With obvious adjustments in part-of-speech classes, subrules (16.1)-(16.6) and (16.8)-(16.11) can serve as a model for the rules needed.

A similar situation holds for the treatment of these constructions in

RG2, where the C_3 constituent of such rules results in an RP ("numeral noun phrase").

7. Adjective-Noun Constructions

Three possibilities must be considered: (a) preposed adjectives governing a noun, (b) preposed or postposed adjectives agreeing with a noun, and (c) predications consisting of nouns and postposed adjectives, both in the nominative case.

RG2 rules

RG2 rules cover all three possibilities.

a. Adjectival government. By a rule analogous to (47), an adjective governing a noun results in an adjective. Only a simple adjective is allowed to govern.[51]

```
(48)  A CP/Y GC/X
      + N CP/X
      = A CP/Y GC/1-X ETC/1
```

Since "noun phrases" (NP), "numeral noun phrases" (RP), "adjective-noun phrases" (ANP) (discussed immediately below) and some other noun-equivalent constituents[52] are produced by RG2 rules, however, the actual number of subrules of RG2 repeating the conditions in (48) is considerable.

Postposition of governing adjectives was not considered because it is generally avoided in modern expository writing.

b. Adjective-noun agreement. Agreement of preposed adjectives was enforced by rules similar to those discussed for certain cardinal numeral constructions and appositions.

```
(49)  A CP/X NG/Y
      + N CP/X NG/Y
      = ANP CP/X NG/Y ETC/2
```

Comment: An adjective agreeing with a noun in case, number, and gender results in an adjective-noun phrase (ANP).

Since accumulative adjective strings resulted in the constituent AP, an AP+ N rule repeating the requirements of (49) was created. Redundant analyses were avoided by not allowing "noun phrases" (NP -- where a noun governs another noun), adjective-noun phrases (ANP), and similar noun-dominated constituents to combine with preposed A or AP constituents.

A rule for postposed adjectives agreeing with a noun is also present in RG2. However. such a rule is destined to create difficulties in analyses unless it is rigidly restricted. For instance, in a string like GOSPLANOV SOHZNYX RESPUBLIK ('state planning commissions of union republics'), the bracketing ((GOSPLANOV SOHZNYX) RESPUBLIK) is incorrect and a rule prohibiting, for instance, a noun phrase containing a right-adjacent adjective to govern a noun in the same case may be a reasonable palliative. However, this question requires additional study along with a related question concerning the ability of a noun to be simultaneously modified by adjectives on both sides.[53]

c. Noun-adjective predications. The ability of postposed adjectives to form a zero-copula predicate when the adjective agrees with a noun is common in all genres of Russian. The following RG2 rule is typical of those developed for such constructions.

```
(50)  N CP/N NG/X
      + A SC/Y-P CP/N NG/X
      = PRED CP/N NG/X ETC/1
```

Comment: A noun in the nominative agreeing with any postposed adjective, other than a pronominal adjective (SC/Y-P, Table II-1), results in a predication (PRED). Pronominal adjectives are excluded because they typically require a dash between them and the noun. A similar rule was provided for the ANP (adjective-noun phrase).

Proposed rules

For the simple instances of adjective-noun constructions described above, the RG2 rules are basically adequate but may require minor adjustments. The more serious problems of adjective-noun agreement arise when one or both of these constituents is coordinative[54] or when, as described next, agreement is affected by the presence of cardinal numerals.

8. Nouns with Cardinal Numerals and Adjectives

Table II-9 summarizes the recommended usage[55] in instances involving numerals in the nominative and inanimate accusative case. In all other cases, adjectives, numerals, and nouns agree in case and number.

RG2 rules

In RG2, combinations of adjectives (A) and cardinal numerals (C) result in the ACP (adjective numeral phrase) constituent, which is produced by both the A + C = ACP and C + A = ACP rules. The ACP constituent acquires the subclass of the numeral and all other tags of the adjective. The tests in each instance are summarized in Table II-10.

Table II-9: Effect of Cardinal Numerals in Nominative and Accusative on Adjective-Noun Agreement

		Case of Adjective (A) as Affected by Position Relative to Numeral (C) and Noun (N)				
		Initial Position	Medial Position		Final Position	
Cardinal		(A C N)	(C A N)		(C N, A,)	
Numeral	Case of		Gender of Noun		Gender of Noun	
Subclass	Noun		feminine	other	feminine	other
SC/S1	noun and numeral in full agreement	(non-standard)	agrees in full with nouns		agrees in full with nouns	
SC/S2 SC/S3	genitive singular	nominative plural	gen. or nom. pl.	gen. pl.	nom. pl.	gen. or nom. pl.
all others	gen. pl.	nom. pl.	gen. pl.	gen. pl.	gen. pl.	gen. pl.

Table II-10: Summary of Test Conditions in RG2 Rules for Cardinal-Numeral-Adjective Constituents (ACP)

Case of Adjective (A) when Numeral (C) is		
	SC/X-S1	
SC/S1	accusative or nominative	other
Agrees in case, number, and gender.	Agrees in case or numeral governs.	Agrees in case.

The subrules for constructions involving ACP and noun constituents did not provide for the nominative plural alternatives shown in Table II-9.

Proposed rules

Possible proposed rules are not provided in view of the fact that in order to present a comprehensive packet, a number of agreement rules would have to be stated which differ little in structural respects from many of the other agreement rules discussed so far.

9. Preposition-Noun Phrases

In RG2, a number of simplifying assumptions had to be made regarding the function of preposition-noun phrases. Generally, a preposition-noun phrase immediately following a simple adjective, adverb, or noun was linked to that constituent. Verbs and predications were allowed to combine with preposition-noun phrases on their left and right. In instances where a preposition-noun phrase occurred between some other potential governor and a verb or a predication, it was linked to the latter. Such necessary but arbitrary assumptions led to obvious consequences.

A study of preposition-noun phrases was undertaken during the contract period. Information contained in the relevant sections of the Academy Grammar and in other sources[56] was coded on special forms and partially classified. Concordances of some of the more frequent prepositions were also studied.

RG2 rules

RG2 rules for the recognition of preposition-noun phrases (PNP) were of the form

(51) P GC/X W/Y
  + N CP/X W/Z
  = PNP SC/Y GC/X W/Z ETC/2

Comment: A preposition (P) governing a noun (GC/X and CP/X on C_1 and C_2, respectively) together with that noun produces a preposition-noun phrase (PNP) whose subclass (SC) is the actual preposition (W/Y and SC/Y on C_1 and C_2 respectively) and whose "government" (GC) is that on the preposition. Since the attribute W is mentioned in the rule, the ETC instruction cannot copy it. Hence, W/Z on C_2 is used to copy the value of W onto C_3.

PNP constituents do not actually govern, but the attribute (GC) was used for another purpose: In the rule N PNP = NPNP, the government of the NPNP (noun-prepositional noun phrase) constituent was taken to be that of the noun less the "government" of the PNP, thereby providing for government across the PNP in a number of instances. For example, PRIEZD V DEREVNH OTQA would be correctly assigned two interpretations: (a) 'father's arrival in the village' and (b) 'arrival in father's village'. However, many counter-examples can be readily found where this rule turns out either to be too permissive or too restrictive.

Several noun phrases were chained together into a PNP-"block" (PNPB) by rules of the form PNP + PNP = PNPB and PNP + PNPB = PNPB. The mechanism of such rules was discussed in Section 2.1.3. The purpose of this construction was to group a string of PNP constituents without attempting to analyze the string further.

Proposed rules

Generally, rules involving PNP constituents are in need of reworking because of recurrent minor inconsistencies. The corrections of the inconsistencies observed are not presented here in detail because of the need for further study of the deeper grammatical problems involved with prepositional phrases.

B. Predications

Although subrules resulting in predications account for about one fifth of the 740 subrules in RG2, only a limited study of predications could be undertaken during the period covered by this report. The complexity of the task becomes apparent when one briefly examines the problems specific to each type of predication. A discussion of types of predications (1) is followed by illustrations of the manner in which predications were treated in RG2 rules (2). A comment about possible new rules (3) concludes this section.

1. Types of Predications

Three types of surface-structure predication constituents can be distinguished: (a) complete predications (subject and predicate are identifiable); (b) subjectless predications (only the predicate is present); and (c) nominative predications (only the subject is present).

Complete predications

These predications, also known as personal predications,[57] consist of a subject and a preoicate. Standard references[58] generally provide a good description of constituent types which can act as subjects in such constructions. Predicates can either be simple (a single finite form of a verb) or can consist of a verb and its complement. Some of the problems of predicates of the latter type are discussed under subclassification of verbs (Section 2. 2. 2), but the question of their structure requires further study. The common instances of predicative (subject-predicate) agreement are equally well described in the same standard references[57] and will therefore not be elaborated upon here. Some of the more difficult problems in predicative agreement include various types of synesis, for example: $3TO BYLA VAJNA4 ZADACA ('this was an important task'), or $BOL6WINSTVO BYLI STUDENTAMI ('the majority were students').[60]

Subjectless predications

The Moscow University Grammar[61] contains a comprehensive discussion of this subject which is not repeated here. It is important to distinguish between personal elliptical predications and impersonal predications. Within the former group, there are three subgroups which are usually referred to as "definite personal" (OPREDEL4H 'I define'); "indefinite personal" ($STALI SOSTAVL4T6 SPISKI '(they) began to compile lists'); and "generalized personal" (typical only of proverbs or belles lettres genres; the second person singular has generalized meaning: $NE POI5EW6, NE NA1DEW6 (literally: 'If thou shalt not seek, thou shalt not find -- One who does not seek does not find (anything)')). The recognition of indefinite personal predications, which are frequent in all styles, is difficult without adequate information regarding the type of subject and objects verb requires and other features discussed in Section 2. 2. 2. For instance, a sentence like $PLANY IZMENILI V PROWLUH P4TNIQU, without restrictions, would be analyzed correctly as 'Plans were changed last Friday' (literally, '(they) changed plans last Friday') and incorrectly as 'Plans changed last Friday'.

Among impersonal predications it is possible to single out several distinct construction types involving the use of verbs, short-form passive participles, infinitives, impersonal predicates in "-O", and negated predicates. As noted above, details can be found in the Moscow University Grammar. A number of points require a brief mention here from the point of view of difficulties created in recognition, however.

When an impersonal predication consists of a verb, it is necessary to distinguish between impersonal verbs and impersonal usage of personal verbs: e.g., SVETAET 'the dawn is breaking' as opposed to GUBY PODERGIVALO 'lips were twitching' (literally: 'something twitched the lips'). (Cf. Section 2. 2. 2.) Since impersonal forms of the latter type, which are third person singular (neuter in the past), are identical to personal forms, they are subject to ambiguity problems similar to those mentioned for indefinite personal predications.

In those instances where the predicate contains a short-form passive participle and an infinitive, detailed rules necessary to obtain correct analyses remain to be worked out. For instance, *REWENIE PRIN4TO POSLAT6, but: REWENIE PRIN4TO POSYLAT6 ('It is the custom to send the decision').

In some constructions involving impersonal predicates in "-O", and some reflexive verbs, dative forms (dative of the "logical subject" as opposed to dative indirect object) create ambiguities. For example: STUDENTAM NEOBXODIMO DOKAZAT6 TEOREMU -- (1) 'Students must prove the theorem', (2) 'It is necessary to prove the theorem for the students'.

Subjectless predications

The Moscow University Grammar[61] contains a comprehensive discussion of this subject which is not repeated here. It is important to distinguish between personal elliptical predications and impersonal predications. Within the former group, there are three subgroups which are usually referred to as "definite personal" (OPREDEL4H 'I define'); "indefinite personal" ($STALI SOSTAVL4T6 SPISKI '(they) began to compile lists'); and "generalized personal" (typical only of proverbs or belles lettres genres; the second person singular has generalized meaning: $NE POI5EW6, NE NA4DEW6 (literally: 'If thou shalt not seek, thou shalt not find -- One who does not seek does not find (anything)')). The recognition of indefinite personal predications, which are frequent in all styles, is difficult without adequate information regarding the type of subject and objects a verb requires and other features discussed in Section 2. 2. 2. For instance, a sentence like $PLANY IZMENILI V PROWLUH P4TNIQU, without restrictions, would be analyzed correctly as 'Plans were changed last Friday' (literally, '(they) changed plans last Friday') and incorrectly as 'Plans changed last Friday'.

Among impersonal predications it is possible to single out several distinct construction types involving the use of verbs, short-form passive participles, infinitives, impersonal predicates in "-O", and negated predicates. As noted above, details can be found in the Moscow University Grammar. A number of points require a brief mention here from the point of view of difficulties created in recognition, however.

When an impersonal predication consists of a verb, it is necessary to distinguish between impersonal verbs and impersonal usage of personal verbs: e.g., SVETAET 'the dawn is breaking' as opposed to GUBY PODERGIVALO 'lips were twitching' (literally: 'something twitched the lips'). (Cf. Section 2. 2. 2.) Since impersonal forms of the latter type, which are third person singular (neuter in the past), are identical to personal forms, they are subject to ambiguity problems similar to those mentioned for indefinite personal predications.

In those instances where the predicate contains a short-form passive participle and an infinitive, detailed rules necessary to obtain correct analyses remain to be worked out. For instance, *REWENIE PRIN4TO POSLAT6, but: REWENIE PRIN4TO POSYLAT6 ('It is the custom to send the decision').

In some constructions involving impersonal predicates in "-O", and some reflexive verbs, dative forms (dative of the "logical subject" as opposed to dative indirect object) create ambiguities. For example: STUDENTAM NEOBXODIMO DOKAZAT6 TEOREMU -- (1) 'Students must prove the theorem', (2) 'It is necessary to prove the theorem for the students'.

The same problem arises with reflexives like POLAGALOS6 ('it was necessary'). In the case of reflexive verbs, additional problems are caused by the relative position of the dative form. For instance: *STUDENTAM* XOTELOS6 POKAZAT6 SVOI ZNANI4 ('Students wanted to show ('felt like showing') their knowledge') and XOTELOS6 POKAZAT6 SVOI ZNANI4 *STUDENTAM* ('One wanted to show one's knowledge before the students').

Nominative predications

This type of predication is encountered in so-called one-word sentences. Three subtypes can be distinguished: (1) existential ($VECER 'evening'; $POJAR! 'fire!'); (2) appellative ($APTEKA 'pharmacy'); (3) demonstrative ($VOT PRIMER 'Here is an example'). With the exception of some of the demonstrative sentences, this type of sentence is almost exclusively found in *belles lettres* and will not, therefore, be discussed further.[62]

2. RG2 Rules

Some two-thirds of the RG2 rules resulting in predications were intended for the recognition of complete predications. Of the remaining third, some twenty rules were intended to recognize instances where predications were combined with other constituents, a comparable number of rules was introduced for impersonal predications, and about five rules were temporary *ad hoc* rules which will not be discussed.

Complete predications

About two-thirds of the rules for the recognition of complete predications involved instances where the predicate was expressed by a finite verb form. The remainder related to predicates whose complements were short forms of adjectives and participles linked by a zero-verb form. Subjects of complete predications in RG2 were either nouns (N), personal pronouns (M), or constructions dominated by them.

In the dictionary, three subclasses of finite verb forms (V) were identified: (a) imperative (SC/M); (b) impersonal (SC/I) and (c) personal (SC/P). Forms of BYT6 ('to be') and 4VL4T6S4 ('to be') coded as AUX (auxiliary) in combination with adverbs (D), infinitives (F), and short forms of adjectives and participles (SF) resulted in finite verb constituents (V) whose tense and person were those of the AUX. Dash (U SC/DASH) in combination with a noun phrase (NP) or a preposition-noun phrase (PNP) resulted in a present tense finite verb form.[63]

The agreement requirements for finite-verb predicate constructions are illustrated in the following rule.

(52) N CP/X NG/Y
+ V CP/X NG/Y
= PRED SC/SV CP/X NG/Y ETC/2

Comment: Since the attribute CP contains both person and case information, agreement in case-person is required. Present tense verbs, because of the number and gender combination in NG, are given all three genders in the singular and the value P in the plural; past tense forms are assigned only the actual gender values. Predication constituents are assigned the attribute subclass (SC) whose values indicate the subject-predicate sequence (SV for this subrule). The rules for combining "verbs" resulting from the combination of a dash with a potential complement are much weaker, requiring only that the noun be in the nominative and the "verb" be of the type mentioned.

In addition, where applicable, rules of the type (52) were followed by government rules of the type (53), which are analogous to other government rules, for instance, (47).

(53) N CP/X
+ V CP/Y GC/X
= NVP CP/Y GC/2-X ETC/2

The NVP "noun-verb phrase" constituent was introduced in part in an attempt to prevent redundant analyses.

The approach employed for recognition of zero-verb form predicates is illustrated in (54).

(54) N CP/N NG/X
+ SF CP/N NG/X
= PRED SC/NSF CP/N NG/X ETC/2

Comment: A noun (N) in the nominative (CP/N) agreeing in number-gender (NG) with a short form of an adjective or participle produces a predication constituent (PRED) of the noun short form subclass (SC/NSF).

Predications combined with other constituents

The most frequent instances involved cases like the two following:

a. In KNIGU BRAT PRODAL ('brother sold the book'), BRAT and PRODAL produce a predication of the type handled by (52). In order to complete the analysis, the predication was allowed to govern a left adjacent noun if the verb was "blocked" by the subject (SC/SV).

(55) N CP/X
+ PRED GC/X SC/SV
= PRED SC/NSV GC/2-X ETC/2

The above rule is similar to (53) in the tests carried out.

b. A similar situation could occur if the example in (a) were preceded by a preposition-noun phrase (PNP): V GORODE KNIGU BRAT PRODAL ('brother sold the book in the city'). Since, as noted in Section 2. 1. 3, a noun was not allowed to pick up a left-adjacent PNP, it was necessary to add a rule similar to (55) where a PRED constituent whose verb was "blocked" by a noun could pick up a left-adjacent PNP constituent.

Impersonal predications

In an attempt to highlight the problems involved in the recognition of impersonal predicates ("words of the category of state"), a special constituent class IP was created consisting of forms in "-O" (e.g., VAJNO 'it is important'). IP constituents produced by RG2 rules resulted from a dictionary item (IP) combining, for instance, an infinitive (F) with a governed constituent. For example, VAJNO POLUCIT6 ('it is important to receive') is analyzed as an IP constituent by a rule of the type IP + F = IP. Since impersonal predicates can govern other constituents on either side, redundant analyses can occur. Such analyses are avoided by employing a device of constituent renaming similar to that described for verbs in Section 2. 1. 3.

Other predications

In an effort to deal with elliptical predications, every finite verb form in the dictionary has been supplied a PRED alternative. The economy of this approach is questionable. The need for supplying PRED alternatives to finite verb forms can be eliminated by merging the PRED and verb constituents into a single class. This possibility was considered, but the change was not carried through because of the need for further study.

3. Proposed Rules

Subject to the elimination of superficial inconsistencies, RG2 rules for recognition of predication constituents provide a basic framework which can be improved by additional restrictions suggested in this section and in Section 2. 2. 2. Given the necessary information, a new or modified set of rules can be written in a relatively short time. However, a considerable amount of time will be required for testing and debugging.

C. Compound Constituents

RG2 contains a number of rules intended for the recognition of compound constituents. These rules have serious limitations which can be eliminated only after a detailed study of coordinative compounds.

RG2 rules

The Russian conjunction I ('and') and the comma have been designated as special constituents: AND and CMA. A given constituent in combination with AND or CMA results in an ANDB (and-block) or CMAB (comma-block) construction whose subclass (SC) is the class name of the constituent with which the AND or CMA combined. For instance, the rules for nouns (N) are as follows.

(56) AND + N = ANDB SC/N ETC/2

(57) CMA + N = CMAB SC/N ETC/2

Several CMAB constructions combine to form a CMAP (comma-phrase) construction by one of the two rules shown below.

(58) CMAB SC/X CP/Y
+ CMAB SC/X CP/Y
= CMAP SC/X CP/Y ETC/1

(59) CMAB SC/PNP
+ CMAB SC/PNP
= CMAP SC/PNP ETC/1

Comment: The two CMAB constructions must agree in subclass (SC) and case-person (CP) in order to be combined into a CMAP construction, except in instances where the CMAB results from a preposition-noun phrase (SC/PNP).

In order to allow for strings consisting of more than two CMAB constructions, the requirements of (58) and (59) are repeated in rules of the form CMAB + CMAP = CMAP. Similarly, CMAB and ANDB result in a CMAP construction, subject to the same restrictions.

To complete the treatment of noun constructions, a noun (N) in combination with a following ANDB, CMAB, or CMAP construction of the noun subclass (SC/N) results in an NPB -- "noun phrase block". The requirements are illustrated in

(60) N SC/X CP/Y
+ ANDB SC/N CP/Y
= NPB SC/X CP/Y ETC/1

The types of rules just discussed work well for the more frequent types of compound expressions, for instance STOL, STUL I KROVAT6 ('table, chair, and bed') type. However, superfluous analyses frequently result because of confusion with detached constituent parts. (Cf. Section 2.1.4D.) Some other problems are brought out in the immediately following discussion of proposed improvements.

Proposed improvements

Coordinative ties are generally poorly described in existing Russian grammars.[64] The essential conditions for the existence of coordinative ties are that (a) the conjuncts in any given construction must perform the same syntactic function and (b) they must refer to different but homogeneous (odnorodnye) concepts. The first point can be illustrated by the following example containing a coordinative subject: IX BESKONECNOE "4 NE ZNAH" I ONI SAMI NAS RAZDRAJALI ('Their endless "I don't know" and and they themselves irritated us'). The second requirement is much more elusive and controversial, and requires additional study.[65]

The RG2 rules rely on a limited set of formal features which seem to be the only ones that can be used at present. Since the attribute CP combines both case and person, the requirements can be stated as follows: all conjuncts in a coordinative compound must be of the same constituent (part-of-speech) class, declined conjuncts must agree in case, and conjugated conjuncts (verbs) must agree in person and number.

Usually, coordinative ties are marked by special coordinative conjunctions[66] which in Russian can be subdivided into three groups according to their patterns of recurrence in individual constructions.

(1) Conjunctions which are typically used only once in a given construction but which can be repeated to signal expressive[67] usage: I ('and'), NO ('but'), ILI ('or'), and some others.

(2) Conjunctions forming constituents used in pairs, or paired conjunctions. For instance, NE TOL6KO . . . , NO I . . . ('not only . . . , but also . . . '). In rare instances, one of the conjunctions can be repeated to lend added expressiveness: NE TOL6KO $IVANOV, NE TOL6KO $PETROV, NO I VSE RABOCIE . . . ('not only Ivanov, not only Petrov, but all of the workers . . . ').

(3) Iterative conjunctions which are used at least twice in a given string. For instance, NI $IVANOV, NI $PETROV, NI $SIDOROV . . . ('neither Ivanov, nor Petrov, nor Sidorov . . . ').

The following four types of coordinative compounds can be considered as being basic for Russian. Their general form, using N to represent a syntactic alternative and CJs, CJp, and CJi to represent single conjunctions, paired conjunctions, and iterative conjunctions, respectively, is as follows.

(a) Asyndetic construction (N, N, N): [ONI] RABOTALI, SOZDAVALI STROILI ('[they] worked, created, built').

(b) Standard construction (N (,) CJs N): OPYTNY1 I ZASLUJENNY1 STOL4R ('an experienced and distinguished joiner').

Note: Both of the above construction types can be extended indefinitely by adding comma-separated N on the left.

(c) Paired construction (CJp N , CJp N): NE TOL6KO PISAT6 NO I CITAT6 ('not only to read but also to write').

(d) Iterative construction (CJi N , CJi N): NI STUL , NI STOL ('neither a chair, nor a table').

Note: This construction can be indefinitely extended by adding (CJi N ,) on the left.

In the following presentation of the general form of coordination rules which can be proposed in place of those of RG2, it is assumed that grammatical ties are satisfied by agreement of CP attributes in each case, but, for purposes of simplicity of exposition, this test is not explicitly shown.

The following notational conventions will be employed in the proposed rules: We shall introduce for conjunctions (CJ) an attribute SUBCL (subclass) with values SINGLE, PAIRED, and ITERAT (iterative) and for the four construction types an attribute CONSTR (construction) with values ASYND (asyndetic), STD (standard), PAIRED (paired[68]), and ITERAT (iterative). Thus, an asyndetic construction is N CONSTR/ASYND. Comma is regarded as belonging to a separate constituent class CMA.

The analysis proceeds as follows.

Step 1. Since the combination of CMA and N results in N CMABL/PLUS (61), and the combination of CJ and N results in N CJBL whose values are copied from the subclass (SUBCL) of the conjunction (62), a single N must not have either of the two attributes (CMABL/*, CJBL/*). In addition, it may not have the attribute CONSTR (CONSTR/*), which is reserved for complete coordinations.

```
(61)  CMA
        + N CONSTR/* CMABL/* CJBL/*
        = N CMABL/PLUS ETC/2

(62)  CJ SUBCL/X
        + N CONSTR/* CMABL/* CJBL/*
        = N CJBL/X ETC/2
```

Because some nouns which have combined with a conjunction must also combine with a comma, the following additional rule is necessary.

```
(63)  CMA
        + N CONSTR/* CMABL/* CJBL/X
        = N CMABL/PLUS CJBL/X ETC/2
```

Step 2. **In order to account for the possibility of single conjunctions and parts of paired conjunctions being repeated, a rule is added to combine successive conjuncts involving them into a single constituent:**

```
(64)  N CONSTR/* CJBL/X-ITERAT CMABL/* NCX1/*
      + N CONSTR/* CJBL/X-ITERAT CMABL/PLUS
      = N CJBL/X NCX1/PLUS ETC/2
```

Comments: This rule does not apply to iterative construction conjuncts (CJBL/X-ITERAT). NCX1 is a locally important tag which is intended to avoid redundant analyses. Note that such repetitions of conjunctions for emphasis must be separated by commas (CMABL/PLUS on C_2).

A similar rule is introduced for combining portions of asyndetic constructions:

```
(65)  N CONSTR/* CJBL/* CMABL/PLUS NCX1/*
      + N CONSTR/* CJBL/* CMABL/PLUS
      = N CMABL/PLUS NCX1/PLUS ETC/1
```

Step 3. Recognition of asyndetic strings can be completed by the following rule.

```
(66)  N CONSTR/* CJBL/* CMABL/*
      + N CONSTR/* CMABL/PLUS CJBL/*
      = N CONSTR/ASYND ETC/2
```

Comment: The C_2 must not have the CJBL attribute (CJBL/*) in order to prevent output of (62) from combining erroneously.

Step 4. Recognition of standard constructions can be completed by the following rule.

```
(67)  N CONSTR/(ASYND) CJBL/* CMABL/*
      + N CONSTR/* CJBL/SINGLE CMABL/(PLUS)
      = N CONSTR/STD ETC/2
```

Comment: The values of CONSTR on C_1 and of CMABL on C_2 can be verbalized as follows: "The attribute can be missing, but if it is not, its value must be the one enclosed in parentheses."

Step 5. Recognition of paired constructions can be completed by the following rule.

```
(68)  N CONSTR/* CJBL/PAIRED CMABL/*
      + N CONSTR/* CJBL/PAIRED CMABL/PLUS
      = N CONSTR/PAIRED ETC/2
```

Comment: Paired conjunction constructions should be additionally differentiated to identify the first and the second half of the pair. This is not taken into account in (68).

Step 6. Recognition of iterative constructions can be completed by the following rule.

(69) N CONSTR/* CJBL/ITERAT CMABL/*
+ N CONSTR/* CJBL/ITERAT CMABL/PLUS
= N CONSTR/ITERAT ETC/2

Comment: In certain iterative constructions, comma is not required, but this possibility is not provided for in the present set of rules.

In the proposed rules considered so far, N has been assumed to be either a constituent spanning a single word or else an endocentric construction. However, in every one of the four basic strings each such simple N can be replaced by an N construction, subject to the restrictions in Table II-11.

Table II-11: Possible N-Constituents in Compound Constructions

	Possible N Constituents				
In the construction	Single N	N CONSTR/ ASYND	N CONSTR/ STD	N CONSTR/ PAIRED	N CONSTR/ ITERAT
Asyndetic	+	-	+	+	-
Standard	+	+(1)	+(2)	+	+(2)
Paired	+	+	+	-	-
Iterative	+	-	+(2)	-	-

(1) Seems generally possible except for instances where the conjunction used in the higher-order construction requires punctuation.

(2) The same conjunction cannot be used to form the N construction and to combine it with other members of the higher-order construction.

Table II-11 provides a general framework which can be used in recognition of coordinative constructions. Minor adjustments may be necessary for nouns and adjectives which can form accumulative strings not considered here. Information presented in Table II-11 can be incorporated without difficulty into the types of rules shown in (61)-(69). However, since some twenty-eight rules would be required, they are not presented.

Because the information presented in Table II-11 could only be hand-tested, it is difficult to establish which of the constructions are actually possible. Normally, the constructions are symmetrical, i.e., the same N-constituent is used throughout. For instance, $ON CITAL $DOSTOEVSKOGO I $BAL6ZAKA, $TURGENEVA I $TOLSTOGO ('He read Dostoyevsky and Balzac, Turgenev and Tolstoy') contains an asyndetic compound of "standard" coordinative constructions. However, assymetry is also possible: $E4 NRAVILIS6 KRASNYE ILI ZELENYE, NO NE CERNYE LENTY ('She liked red or green ribbons but not black ones') where a "standard" coordinative compound consists of a "standard" coordinative construction and a single N. One of the major difficulties in the study of coordinative compounds is that there are a variety of options, with the choice among them strongly affected by stylistic considerations. Moreover, in addition to developing restrictions based on semantic compatibility (cf. footnote 65), it is necessary to consider the syntactic function of conjuncts. For instance, the sentence NE STOL6KO BRAT SKOL6KO OTEQ PROCITAL KNIGU ('not so much the brother as the father read the book'), where the paired conjunction is in the subject and hence cannot exhibit its additional adverbial properties, is at best marginal. However, ON BYL NE STOL6KO BRATOM, SKOL6KO OTQOM ('he was more of a father than a brother'), where the same paired conjunction occurs in the predicate, is standard usage.

D Punctuationally Delimited Constituents

As noted in Section 2.1.2, RG2 includes a small number of rules intended for the recognition of subordinate clauses and detached constituents. In text, both constituent types are set off by detaching punctuation, the recognition of which involves a number of difficulties, some of which are described below.

Detaching punctuation

Detaching punctuation is usually in the form of paired commas which are sometimes replaced by dashes or a combination of a comma and a dash.[69] For a string B which is dominated by a string A and detached, the expected pattern of punctuation is (A, B,). However, in sentence-final position, the same pattern is of the form (A, B.) and may be difficult to distinguish from an asyndetic coordinative compound. For instance, $MY VIDELI DVUX PTIQ, DVUX UTOK may be interpreted as either 'We saw two birds, (i.e.) two ducks' or 'We saw two birds (and) two ducks'. Similar problems can occur in sentence-initial position: $T4JELO BOL6NYE, OFIQERY I SOLDATY NAPRAVL4LIS6 V TYL. This sentence has two possible interpretations: (a) 'Being seriously ill, the officers and soldiers were sent to the rear' or (b) 'The seriously ill, the officers, and the soldiers were sent to the rear'.

Instances where a dash and a comma are used in combination can be even more confusing. For example, $JITELI GORODA - MUJCINY I JENSINY, ODETYE PO-PRAZDNICNOMU, OJIDALI PRIEZDA GOSTE1 can be interpreted correctly as (a) 'Inhabitants of the city -- men and women dressed in their Sunday best -- awaited the arrival of visitors' or incorrectly (b) 'The inhabitants of the city are men, and women dressed in their Sunday best awaited the arrival of visitors'.[70]

Related to the problems involving boundary recognition is the problem of "bridging" detached constituents. For instance, in $ONI PELI, VY1D4 NA SQENU, PESNI I ROMANSY, in addition to the correct interpretation, (a) 'Having entered the stage, they sang songs and romances', an incorrect interpretation, (b) 'Having entered the stage, the songs and the romances, they sang', is difficult to avoid.

Finally, a recurrent problem is that both clauses and detached constituents can form coordinative compounds, in which case the punctuation may be dramatically different from what would normally be expected: e.g., $MY GUL4LI I KOGDA SVETILO SOLNQE I KOGDA WEL DOJD6, 'We took walks both when the sun was shining and when it rained', where the comma that would normally precede each KOGDA clause is omitted.

Subordinate clauses

In RG2 rules, subordinate clauses are treated on a token basis essentially limited to KOTORY1-relative clauses and CTO- and CTOBY-clauses. The latter two constructions are further limited to instances where they are governed by either a noun or a verb (e.g., NADEHS6, CTO ... ('I hope that ...')). KOTORY1-clauses created difficulties when the relative pronoun was "buried" in the clause: ... CELOVEK, NA USTALOM LIQE KOTOROGO BYLA ZAMETNA ULYBKA, ... ('...a man on whose tired face one could notice a smile...'). Some methods for coping with this problem have been considered but not actually implemented: gender, number, and animateness of the pronoun can be recorded in a special tag and this tag carried as the attribute of the clause.

In addition to problems of formal agreement, future research must take into account various selection restrictions on combinations of relational words (relative pronouns) with other constituents. For instance, since in ... KOMANDIR GARNIZONA, KOTORY1 BYL JENAT NA... ('... commander of the garrison, who was married to ...'), KOTORY1 is the subject of a verb requiring an animate subject, it can refer only to an animate noun.

The subordinating conjunctions were also investigated during the contract period. However, the results are inconclusive. Such important information as the relative position of the subordinate clause with respect to the main clause, the ability or inability of a relative clause to refer to a single word or act as a sentential modifier, tense agreement between the verbs in both the main and the subordinate clauses and many other problems are dealt with rather superficially in the Academy Grammar[71] and will require additional study.

Detached constituents

The attempts to deal with detached constituents have been generally satisfactory in the most obvious instances, i.e., when an adjective or a gerund is in sentence-medial position and is clearly delimited by punctuation which can be absorbed into the constituent. Because the large number of rules that would have been required to handle them adequately in RG2 were not included, detached constituents in sentence-final position were often incorrectly identified only as parts of coordinative compounds, with which they are superficially identical. Moreover, because of the absence of any definitive study on the subject, restrictions on the ability of a constituent containing a detached constituent to combine with other constituents were not worked out in a satisfactory way.

E. The CSA Syntactic Dictionary

In order to conduct tests on the 160-sentence sample of Pravda editorials, an experimental dictionary of about 1650 Russian word forms and punctuation symbols, covering the vocabulary and punctuation in the sample, was compiled. Linguistic coding was completed for a similar full-form dictionary covering the entire 1600-sentence corpus of Pravda editorials. However, these additional entries were not incorporated into the CSA dictionary because the machine processing necessary to perform this operation was not completed.

The entries in the CSA dictionary consisted of the Russian word forms or punctuation symbols followed by their respective syntactic alternatives. For instance, the entry for the form GLUBOKO1 ('deep') was:

(70) *GLUBOKO1

1 A SC/A CP/G NG/F W/GLUBOKO1

2 A SC/A CP/D NG/F W/GLUBOKO1

3 A SC/A CP/I NG/F W/GLUBOKO1

4 A SC/A CP/L NG/F W/GLUBOKO1

Comment: GLUBOKO1 is classified as an adjective (A) whose subclass (SC) indicates that it is a positive degree adjective proper (A) (see Table II-1). The attribute W (word) has the actual word form as its value (W/GLUBOKO1). All of the alternatives are feminine singular

(NG/F) adjectives. (For the meaning of the attribute NG, see Section 2.1.3.) The alternatives differ in the value of the case-person (CP) attribute (cf. 2.1.3) which has values genitive (G), dative (D), instrumental (I), and locative (L).

The entry for the dash (--) was as follows:

(71) * --

1 U SC/DSH W/DASH

Comment: In RG2, all punctuation symbols, with the exception of comma, are assigned to the constituent class U (U). The SC/DSH is the subclass "dash" of U. The tag W contains the actual name of the symbol.

A detailed description of the constituent classes and tags appearing in the CSA dictionary is not provided here because the pertinent information has already been presented in the discussion of selected rules of the CSA Russian grammar.

2.2 Subclassification studies

2.2.1 Nouns

A pilot study of possible subclassification of nouns was conducted with the aid of Library of Congress personnel. This study was an outgrowth of work performed under earlier contracts[72] and was based, in part, on semantic subclasses of nouns encountered in the course of work on preposition-noun phrases described in A9 of Section 2.1.4 of this report.

In an attempt to develop uniform standards of classification to facilitate eventual processing of the results by computers, a special questionnaire (Figure II-2) was compiled to be used in conjunction with classification charts. The questionnaire contained yes-no questions and room for recording the codes of information derived from the charts. Whereas the queries in the questionnaire were directed at exploring the analyst's Sprachgefühl and were unstructured, the charts represented an attempt to illustrate graphically a proposed system of classification in the form of a branching diagram. It was soon discovered, however, that the charts were incomplete and difficult to follow. As a consequence, the questionnaire was revised and used alone in subsequent experiments.

This section contains a description of the questionnaire, followed by an assessment of the results obtained in the pilot study. It concludes with some interim proposals for noun classification.

Figure II-2: TEST VERSION OF NOUN CLASSIFICATION SHEET REVISED January 14, 1966

Miscellaneous	Yes	No	
Test the frame 4 PROWEL + N_{acc}			74
4 PROWEL + N_{inst}			75
PROPER NAME			77
NONL SCRIPT			78
N_x has shape			79
PRIKOSNULIS6 K			83
N_x governs: DATIVE			84
INFINITIVES			85
O-CLAUSES (OTNOSITEL6NO)			86
Enter codes from the diagram.			
Is there more than one code?			87
If more than one code, rank in preference: 1.______in cols______ 2.______in cols______ 3.______in cols______ 4.______in cols______			88 ---- 100
3LEKTROPROVODNOST3 RUSSIAN WORD ELECTRIC CONDUCTIVITY ENGLISH MEANING (to serve as a guide)			
Indicate here if there are any comments, then use the back of the sheet, if necessary.			

N_x in genitive	Yes	No	
MNOGO+N_{sing}			36
MNOGO+N_{plur}			37
PO SLUCAH+event			38
VO VREM4			39
VNUTRI			41
SOSTO4NIE			42
DVADQAT6			43
POST, MESTO			44
ISKLHCENIE or SOKRA5ENIE			45
= subject gen.			46
= object gen.			47
N_x is in acc. 4 CUVSTVUH			48
4 PRIWEL V+			50
4 SIDEL			51
Instrumental			
4 STRADAL ILI BOLEL			52
NAZNACEN...			53
...by N_x...			54
...as N_x...			55
.....N_x...			56
other meanings			57
Can N_x be perceived by sense of sight			58
hearing			59
touch			60
taste			61
smell			62
N_x is in gentv. KUSOK, CAST6			63
PREDSTAVITEL6			64
NACAL6NIK, DIREKTOR, KOMANDIR			65
IMPERS. PREDIC. (semantic meaning)			72

	"A"	"B"	AB	A	B	0	
Nx acts as subject..........	BOLIT	BOLEET					1
	IDET	XODIT					2
	LETIT, LETAET by its own power						3
	TECET ('flow') PROTEKAET	TECET ('leak') PROTEKAET					4
	PRODOLJ-64	NACALS4					5
	PADAL, UPAL						6
	RASTET	UMEN6W-S4					7
	VSKOCIL PODPRYGNUL						8
	CUVSTVUET, NERVNICAET						9
	POGIB						10
..........Subject to proper agreement..........	Test ability to combine...						
	VYSOKI ('tall')	VYSOKI1 ('high')					11
	TONKI1 ('thin')	TONKI1 ('refind')					12
	KRUPNY1 ('big')	KRUPNY1 ('great')					13
	GLUBOKI1 ('deep')	GLUBOKI1 ('profound')					14
	MELKI1 ('fine')	MELKI1 ('petty')					15
		PON4TNY1					16
	PLOSKI1	KRUGLY1					17
	DOLGI1	DLINNY1					18
	KRATKI1	KOROTKI1					19
	TVERDY1	M4GKI1					20
	JIDKI1	GUSTOI					21
	WIROKI1	UZKI1					22
	TEMNY1	4SNY1					23
	TIXI1	GROMKI1					24
	GOR4CI1	XOLODNY1					25
	ZASTENCIVY1	VESELY1					26
	BYSTRY1	SIL6NY1					27
	BESQVETNY1	MUCITEL6NY1					28

DATE

INITIALS

The questionnaire

The questionnaire was developed at IBM Research, but the bulk of the experimental classification work was done by the Library of Congress lexicographers. The numbering of questions is not continuous because of the revisions made.

The English meanings in the questionnaire were intended to serve as a guide for the identification of Russian words (see lower left corner of Figure II-2). Non-standard or archaic usage was not considered. Reasons for including a given question or group of questions are outlined below. Since the design of the questionnaire is of a tentative nature, accompanying each explanation are comments on how the questions were in fact interpreted by the analysts taking part in the classification of nouns.

Questions 1-10: This group of questions is intended to establish the properties of the noun under analysis (N_x) which are manifested in its function as the subject of the sample verbs. Thus, BOLIT ('it aches') excludes any animate agents and is limited to the nouns denoting parts and certain defects of the body (RUKA 'hand', RANA 'wound'). BOLEET ('is sick, ailing') typically admits animate agents and can also be combined with (inanimate) nouns denoting living organisms (DEREVO BOLEET 'the tree is ailing'), while CUVSTVUET ('senses') and NERVNICAET ('is nervous') admit only animate agents. The notion of an animate agent includes both the grammatically animate nouns (CELOVEK 'man') and formally inanimate nouns characterized by "real" animateness (NASELENIE 'population'). To distinguish the two types of animateness, RASTET ('grows') and UMEN6WAETS4 ('diminishes') were added, since the former admits both types of animate agents and the latter excludes grammatically animate nouns and a few collective nouns of the type CETA ('couple'). Hence, on the basis of answers provided regarding possible combinations with the above-mentioned verbs, several semantic and syntactic features of a noun can be quickly established.

Questions 11-28: Ability-inability of the N_x to combine, subject to proper agreement, with the sample adjectives is intended to bring out additional semantic nuances.

Questions 36-47: In all these tests the noun being analyzed must be in the genitive case.

Questions 36-37: Ability-inability to combine with adverbial quantifiers of the type MNOGO ('much, many') and the number of the governed noun are important formal features. Some additional distinctions are: All count nouns combine with MNOGO in genitive plural (MNOGO STUL6EV 'many chairs'). However, many nouns which combine with MNOGO in genitive plural are not countable (MNOGO TRUDNOSTE1 'many difficulties'). The so-called "mass" nouns combine with MNOGO in genitive singular (MNOGO SAXARA 'much sugar'). However, the same is true of many other nouns, among them WUM ('noise'), ZLO ('evil'), CVANSTVO ('boastfulness'). Many collective nouns of the type PROLETARIAT ('proletariat') or STUDENCESTVO ('students - collectively') cannot combine with MNOGO at all. The same is true of most "abstract" nouns, especially nomena actionis like ZAROJDENIE ('nascense'), VOPLO5ENIE ('incarnation') and others.

Questions 38-41: Ability to combine with prepositions and prepositional constructs is self-explanatory. The semantic criteria sought are suggested by the meaning of the respective prepositions: PO SLUCAH ('on the occasion of'), VO VREM4 ('during'), and VNUTRI ('inside'). VO VREM4 should not be confused with VO VREMENA ('in the days of'), nor VNUTRI with V ('in').

Question 42: Is N_x governed by SOSTO4NIE ('the state of')?

Question 43: This question should be answered "yes" only for normally countable nouns (*DVADQAT6 $GERMANI1 'twenty Germanies').

Question 44: Can N_x appear in the context $ON NAZNACEN NA POST, MESTO ... ('he is appointed (to the post of) ... ')? The question should be answered "yes" only when the resultant construction has equivalents like the following: $ON NAZNACEN NA POST DIREKTORA - $ON NAZNACEN DIREKTOROM ('he is appointed director') or $ON NAZNACEN NA DIREKTORSKI1 POST ('he is appointed to the office of a director'). These additional restrictions are intended to eliminate the instances of the genitive only denoting "possession": $ON NAZNACEN NA MESTO BRATA ('he is appointed to his brother's office').

Questions 45-47: Using ISKLHCENIE ('expulsion, elimination') or SOKRA5ENIE ('contraction, reduction') as representing a class of deverbal nouns, the question concerning subject genitive-object genitive distinction was intended to confirm animate-inanimate distinction. It was initially assumed that animate nouns could be both subject and object genitives, but inanimate nouns only object genitives. This was an unfortunate oversight. For example, in VRA5ENIE LUNY ('rotation of the moon'), one obviously is confronted with subject genitive (Cf. LUNA VRA5AETS4 'the moon rotates'). The questions should have been eliminated.

Questions 48-51: In these tests the noun being analyzed should be in the accusative. $4 CUVSTVUH (question 48) should be used in the sense 'I experience' or 'I sense' (Cf. Russian $4 ISPYTYVAH CUVSTVO ... ($N_{genitive}$)). Only nouns denoting various emotions and sensations seem to fit in this context; for instance, BOL6 ('pain'), NEGODOVANIE ('disgust'), RADOST6 ('joy') and some others. $4 PRIWEL V (question 50) can combine with nouns denoting points in time (POLDEN6 'noon') or locations in space (GOROD 'city'), some of which can best be called generalized locations. Instances of the latter type include military units such as POLK ('regiment'), CAST6 ('unit'), ROTA ('company'), and others. In these cases PRIWEL is a verb of motion: $4 PRIWEL V P4TNIQU ('I came on Friday'). When combined with abstract nouns denoting certain emotional states, PRIWEL becomes a link verb as evidenced in part by English translations: $4 PRIWEL V UJAS ('I was horrified'); $4 PRIWEL V ISSTUPLENIE ('I was outraged'). $4 SIDEL ('I sat') in question 51 can be followed only by the accusatives of time (e.g., GOD 'year', CAS 'hour', NEDELH 'week'). The three frames, then, help to identify a number of divergent semantic distinctions: locations, time periods and points in time, sensations, emotions, and emotional states.

Question 52: $4 STRADAL ILI BOLEL ('I suffered from or was sick with') can be followed by a noun in the instrumental singular denoting afflictions and diseases ($4 BOLEL TIFOM 'I was sick with typhoid'), or time periods in the same way as questions 53 and 75. Plural nouns which also can follow $4 STRADAL ILI BOLEL are restricted to countable nouns, namely, those which can follow $4 SIDEL (question 51).

Questions 53-57: The next four questions represent an attempt to define the function in the instrumental[73] of nouns which can combine with NAZNACEN ('appointed') (question 53). Thus, "by" (question 54) is intended to signal the agent instrumental ($ON NAZNACEN $IVANOM 'He was appointed by Ivan'; $IVAN EGO NAZNACIL'Ivan appointed him'); the use of "as" (question 55) is equivalent to 'when he was' (NAZNACEN MOLODYM CELOVEKOM 'appointed as a young man'). Question 56 probes the possible use of the N_x as a complement of the type NAZNACEN KOMANDIROM ('appointed as a commander'). Other meanings include various adverbial functions like NAZNACEN PRIKAZOM in the sense of NAZNACEN PO PRIKAZU ('appointed on the order').

Questions 58-62: Questions 58-62 are self-explanatory but have proved difficult to answer. Basically, they represent an attempt to identify "concreteness" in the literal sense.

Questions 63-65: The three questions require N_x to be in the genitive. KUSOK, CAST6 ('piece, part, portion') (question 63) establish "divisibility" of the N_x. It seems that only literally "abstract" nouns cannot follow one of the nouns of KUSOK or CAST6-type. For instance, *CAST6 GOTOVNOSTI ('a portion of preparedness'). Ability to combine with PREDSTAVITEL6 ('representative') (question 64) is restricted (a) to nouns denoting beings (PREDSTAVITEL6 RABOCIX 'workers' representative') and to formally inanimate nouns which may acquire "real" animateness (e.g., ZAVOD 'factory' or POLK 'regiment'); (b) to nouns denoting genera (PREDSTAVITEL6 MLEKOPITAH5IX 'a representative of the mammals'); and (c) to individual humans mentioned by name. NACAL6NIK ('head'), DIREKTOR ('director'), and KOMANDIR ('commander') should help to single out nouns mentioned in (a).

Question 72: This question establishes whether or not N_x has a corresponding impersonal predicate. For instance, STRAX-STRAWNO ('fright, fear -- it is frightening'); OPASNOST6-OPASNO ('danger--it is dangerous'); MOROZ-MOROZNO ('cold, frost -- it is chilly (freezing)'); and others. It seems that this test would apply almost exclusively to nouns denoting certain states, including meteorological conditions (TUMAN 'fog' or JARA 'heat'), and emotions.

Questions 74-75: Ability to combine with the verb in both frames is intended to single out nouns denoting time periods and locations. $4 PROWEL followed by the N_x in the accusative usually corresponds to the construction known as accusativus extensionis (accusative of extent): $4 PROWEL DVADQAT6 MIL6 ('I went twenty miles'). Constructions with the instrumental can either contain nouns denoting locations (Cf. Latin ablativus loci): $4 PROWEL POLEM ('I went via the field'), or those functioning as time adverbials: $4 PROWEL LETOM ('I went through in the summer'). Unfortunately, many other constructions can occur in both frames, especially if the context is increased. For instance, when the noun is in the accusative case, PROWEL can combine with nouns denoting school subject course names and other nouns: $4 PROWEL KURS NAUK ('I completed a course of study'). A noun in the instrumental can perform the same function as the one possible in question 55. For instance, MOLODYM OFIQEROM $4 PROWEL KURS NAUK ('as a young officer I completed a course of study').

Question 77: Is N_x a proper name?

Question 78: Is N_x a "nondescript" noun? This proved to be one of the more difficult questions and, based on experience to date, this category should be reworked or eliminated. The category was adopted from Eaton (1961; 437) and was intended to identify nouns denoting items characterized only by shape (SPIRAL6 'spiral'), function (CISLO 'number'), or quantity (U1MA 'an awful lot'), etc.

Question 79: The ability to have a (geometric) shape was intended as a confirmation of concreteness of the noun.

Question 83: The ability to combine with PRIKOSNUT6S4 K ('to touch') was intended to reinforce the test for concreteness described immediately above.

Questions 84-86: Ability to govern the dative, infinitives, and deliberative object (O 'about'; OTNOSITEL6NO 'concerning') phrases was queried.

Questions 87-100: These locations were reserved for codes obtained from charts. As noted in the introduction to this section, charts were not used after limited initial testing.

Results of the pilot study

A total of about 2, 000 nouns were processed in the course of the study. As expected, considerable difficulties were encountered in a number of areas. For instance, many of the component questions of the questionnaire were subject to different individual interpretations and it was difficult to establish uniformity. Quite often several meanings of the same word form tended to become superimposed or one another despite apparent efforts to the contrary. Although extensive, the questionnaire was weak in distinguishing varieties of abstract and deverbal nouns of the type UPRUGOST6 ('elasticity') or PROXOJDENIE ('passage'). Hence it appears that several specialized questionnaires would have to be developed.

Because of these difficulties, several methods of computer and manual sorting generally failed to group together items which might be expected to be encoded identically. Since the answers contained in each questionnaire, when reduced to numerical codes, were of the appropriate form (numerical vectors), several attempts were made to process samples of nouns with the aid of pattern recognition programs operating on such data which were developed for another purpose within IBM Research. The use of these programs required that an arbitrary number of groups had to be specified in advance. By means of clustering techniques analogous to those described in Casey and Nagy (1966; 95-6) the pattern recognition programs divided a given sample of nouns into the preassigned number of groups. Although a number of questions can be raised concerning the validity and the methodology of this procedure, the results obtained are interesting. Predictably, sufficiently close groups or portions of larger groups brought together by the pattern recognition programs reflected quite clearly such categories as animate nouns or mass nouns as, for example, the following: GRAD ('hail'), KREMNI1 ('silicon'), KRISTALL ('crystal'), PLENKA ('film'), SERA ('sulfur'), and POROX ('powder') -- all of which can be called in some sense "mass nouns".

One example will serve to summarize the types of problems encountered in trying to achieve uniformity of coding. The animate nouns in Table II-12 below brought together as a group by the pattern recognition program differed as shown with regard to their encoding on the questionnaire. Many of the divergent answers shown in Table II-12 can be explained by legitimate differences of opinion, others are outright errors. Whatever the causes, the effect on processing the data by a computer is obvious.

The pilot study described above produced few immediately usable results. However, it provided some valuable insights into the possibility of

Table II-12: Range of Questionnaire Responses for Some Representative Animate Nouns

QUESTION		NOUNS						
(No.)	Comment	CLEN ('member')	GLAVA ('head')	OPERATOR ('operator')	PASSAJIR ('passenger')	RABOTNIK ('worker')	UCENY1 ('scientist')	SPUTNIK ('companion')
(1)	Can N_x be the subject of BOLEET ('is ill')?	-	-	+	-	+	+	-
(10)	Can N_x be the subject of POGIB ('perished')?	+	-	-	+	+	-	-
(11)	Can N_x be modified by VYSOKI1 ('tall')?	+	-	-	-	+	+	-
(13)	Can N_x be modified by KRYPNY1 ('great')?	+	-	+	-	+	+	-
(15)	Can N_x be modified by MELKI1 ('petty')?	+	-	-	-	-	-	-
(24)	Can N_x be modified by TIXI1 ('quiet')?	+	-	+	+	+	+	+
	Can N_x be modified by GROMKI1 ('loud')?	+	-	-	+	-	-	-
(42)	Can N_x in the genitive follow SOSTO4NIE ('state of')?	+	-	+	+	+	+	-
(45)	Can N_x combine with ISKLHCENIE ('expulsion, exclusion'), etc. ?	+	-	+	+	+	+	-
(46)	Is this a subject genitive?	+	-	+	-	+	+	-
(47)	Is this an object genitive?	+	-	+	+	+	+	-
(64)	Can N_x in the genitive follow PREDSTAVITEL6 ('representative of')?	-	+	-	-	+	-	-
(65)	Can N_x in the genitive follow NACAL6NIK ('head'); DIREKTOR ('director'), etc. ?	-	-	+	-	+	-	-

using such methods in the future. Some short-range prospects are described immediately below.

Some interim proposals

A comprehensive classification of nouns suitable for syntactic analysis of Russian is not a likely prospect for the near future. It seems that in experimental research on Russian grammar one should concentrate on specific syntactic and semantic features of nouns. Such features can be gradually introduced into the recognition system which would provide a vehicle for testing them and evaluating their effectiveness. In addition to features proposed in this report for the recognition of appositive constructions (cf. A4 of Section 2. 1. 4), the following seem to merit further consideration.

a. Animateness. Formally, animateness is reflected in identity of form of the genitive and accusative case of certain nouns (the notable exceptions are feminine nouns of the first declension in -A (RUKA 'hand') and those of the MAT6 ('mother') type). As suggested in the discussion of questions 1-10 of the questionnaire, however, it is necessary to consider a category of animate agents which can act as animate agents of certain verbs although formally inanimate. For instance, KOLXOZ SLUWAL . . . ('The collective farm listened to . . . ').

b. Personification. Ability of a noun to act as an animate agent should perhaps be treated as a part of the broader problem of personification,[74] i.e., the ability of an abstract noun to act as a concrete agent: $EGO SLOVA VREZALIS6 V PAM4T6 (literally: 'His words plowed into the memory'). Related to but not necessarily identical with personification are problems of "concretization" of abstract nouns as evidenced from such factors as countability and pluralizability. For instance, SOKRA5ENIE RASXODOV ('curtailment of expenditures'), but DVADQAT6 WEST6 SOKRA5ENI1 ('twenty-six abbreviations').

c. Substantivization. Although substantivization (ability of a norphological part of speech other than a noun or of other constructions to function syntactically as a noun)is not normally considered a part of noun subclassification, such substantivized items exhibit a number of peculiar features. For instance, the ability to combine with adverbs (T4JELO BOL6NO1 'seriously ill patient') or the ability to be modified by adjectives (BESKONECHNYE "NEL6Z4" VREZALIS6 V PAM4T6 'The endless "you can't do that"'s plowed into the memory').

d. Deverbal nouns. Since deverbal nouns, typically those having the -ENIE, -KA, and zero-affix (VZRYV 'explosion'), duplicate many of the characteristics of the verb, they should be identified as such. This feature is important in a variety of constructions but especially in preposition-noun phrases and instances when a deverbal noun governs a noun in the genitive. In preposition-noun phrases, the different syntactic function of the construction for normal and deverbal nouns is suggested by the contrasts in translation: ON PRIWEL V KOMNATU ('he came into the room'), versus ON PRIWEL V VOLNENIE ('he became excited); ON JIL PRI BOL6NIQE ('he lived at the hospital'), versus ON POSTRADAL PRI POSADKE ('he suffered injury in landing'). In instances when the deverbal noun governs a noun in the genitive, object genitive and subject genitive constructions are distinguished on the basis of the underlying verb. Thus if KRICAT6 ('to scream') is an intransitive verb , KRIK KOMANDIRA ('commander's scream') can only be a subject genitive; if NERVNICAT6 ('to be nervous') can only have animate subjects, it is possible to encounter NERVNICANIE JENY ('wife's nervousness'), but not *NERVNICANIE STULA (*'nervousness of the chair').[75] Within the group of deverbal nouns one may further distinguish between those that preserve the verbal characteristics and those that remain deverbal in derivation only: PROVODY ('seeing off (of someone)') versus PROVODA ('wires') or SOKRA5ENIE ('(process of) curtailment/contraction') versus SOKRA5ENIE ('abbreviation/abbreviated word').

e. Action nouns. A number of nouns denoting various actions are not deverbal in the strict morphological sense as those mentioned above (SPIN 'spin', REAKQI4 'reaction') or their morphological relationship to a verb is obscure (RABOTA 'work'). Such nouns share certain properties with deverbal nouns but require additional study.

f. Adverbial function. Many nouns, especially in the accusative and instrumental (cf. discussion of questions 74 and 75 of the questionnaire), acquire adverbial functions: BOLVANKA VESIT TONNU ('the ingot weighs a ton').

The list of such features can be extended. Some have been suggested elsewhere,[72] others can be easily derived in the course of research once it is undertaken. In most instances, however, such a classification will yield little without a corresponding parallel analysis of other parts of speech -- in particular, verbs.

2. 2. 2 Verbs

A limited study of verbs was undertaken, utilizing a portion of the classificatory criteria described in Andreyewsky (1965). Some 1800 verbs contained in a special brochure published by the USSR Academy of Sciences (Demidova et al., 1963) were analyzed jointly at IBM Research and at the Library of Congress. As with the nouns, the results of the verb classification were processed by the pattern recognition programs.

The test

The verbs were coded according to their ability to combine with selected prepositional phrases, certain adverbs, and the CTO-introduced object clauses. The objective was to test not only the ability of a given verb to co-occur with certain types of phrases or classes of adverbs, but also to trace what effect, if any, the verb has on their syntactic function. Participants in the test were presented with a list of potential verbal environments (Figure II-3) and asked to indicate whether or not the verb could occur in them in one or more of the indicated senses. The particular nouns and

Figure II-3: TEST ENVIRONMENTS OF VERBS

1) ...DO MEN4
(A) before me
(B) as far as me

2) ...DO RASSVETA
(A) before dawn
(B) until dawn

3) ...IZ-ZA STOLA
(A) because of the table
(B) from behind the table

4) ...K MITINGU
(A) for the meeting
(B) to the meeting

5) ...K NAM
(A) to us
(B) toward us

6) ...ZA OBEDOM
(A) after (to get) dinner
(B) during dinner

7) ...U $ZINY
(A) at Zina's
(B) from Zina

8) ...POD KAPUSTU
(A) for cabbage
(B) under cabbage

9) ...ZA STOL
(A) at the table
(B) behind the table

10) ...ZA BRATA
(A) in brother's place
(B) for brother's sake

11) ...PO OWIBKE
(A) a mistake apiece
(B) by mistake

12) ...45IK IZ-POD UGL4
(A) coal crate
(B) crate from under the coal

13) ...O STOL *
against the table

14) ...PO VODU *
to get water

15) ...CTO NAPIWET *
that + (subject) + will write

16) ...NADVOE *
in two (as in cutting)

17) ...OCEN6 *
very much

18) ...O SESTRE *
about the sister

* only test ability to combine in the meaning indicated

pronouns appearing as preposition complements in the sample environments were regarded as representatives of classes rather than in terms of their precise lexical content, e.g., RASSVETA ('dawn') in item (2) of Figure II-3 stood for the class of "event nouns".

Results of the test

An analysis of the results obtained from this study of verbs suggests that although the criteria used can improve the recognition of verb-governed preposition-noun phrases, a much more comprehensive study of verbs will have to be undertaken in the future. Some of the questions can be formulated much more clearly by substituting interrogative adverbs or other prepositional constructions. For instance, 3A and 3B can be replaced by POCEMU ('why') and OTKUDA ('where... from'), respectively.

Possible extensions of verb classification

The limited study of verb classification conducted during the contract period represented an attempt to deal with a number of the so-called *Aktionsart* distinctions, implicit in the ability of verbs to combine with various adverbials. Briefly, this view holds that while a pair of verbs like PROCITAT6 ('to read', perfective) and PROCITYVAT6 ('to read', imperfective) is an aspectual pair, the distinction in aspect is secondary in the case of a pair such as CITAT6 ('to read', imperfective) and PROCITAT6 ('to read', perfective); instead, one deals with *Aktionsart*, a "manner of action" distinction, because the latter verb form describes a particular way the process of reading transpired, as reflected in the possible English equivalent 'to read through'.[76] Since most *Aktionsart* distinctions are related to various affixed forms of the basic verb, affixation of the Russian verb and *Aktionsart* must be studied concurrently.

Aspect and *Aktionsart* are important for the recognition of constructions consisting of verbs and adverbials. Thus, VSEGDA ('always') only rarely occurs with perfective verbs (*VSEGDA PROCITAL 'always read through'). while the attenuative *Aktionsart* of the verb -- e.g., PODZABYL ('(partially) forgot') -- cannot combine with adverbs like KONCATEL6NO ('completely, finally'). The prefix of a verb is also important in certain instances of verb government of preposition-noun phrases, e.g., *PODOWEL K* ... ('walked up to ...') or *VNES V* ... ('carried into ...'). Some restrictions on the type of noun which can appear in such constructions must be worked out -- for example, *NAPISAL NA* BUMAGE ('wrote on paper'), but NAPISAL NA RADOST6 STRANE ('wrote to the delight of the country').

The need to develop selection restrictions for possible subjects and objects of a given verb has been mentioned in conjunction with noun classification. In addition, it is necessary to establish a more basic distinction:

the ability-inability of a given verb or its forms to appear in personal and impersonal sentences. Thus, NERVNICAT6 ('to be nervous') and SVETAET ('it is getting light') are examples of a personal and an impersonal verb, respectively; KAZALOS6 in NAM KAZALOS6 ('it appeared to us') and PUGALO in NAS PUGALO ('we were afraid') are impersonal forms of personal verbs. The semantic content of the subject is sometimes important in establishing the voice of a predication containing a reflexive verb as shown, for example, by the contrast between DETI MOHTS4 ('children wash themselves') and REDISKA MOETS4 ('radishes are washed').

Although certain features of verbal government are encoded in the Russian Master Dictionary (RMD), additional improvements are necessary. These include (a) distinctions of the sort discussed above under classification of nouns and (b) information about the ability of a transitive verb to function intransitively (in its "absolute form").

Yet another area requiring study is treated in most Soviet grammars under the heading of "compound predicates" (sostavnye skazuemye).[77] Classes of complements have to be defined and the ability of a verb to combine with each class studied. Thus, it seems that almost all Russian verbs can have a full-form adjectival complement in the instrumental or the nominative case: UMER MOLODYM ('died young'). Some verbs can, in addition, have substantive complements (KAZALS4 GENERALOM 'appeared to be a general') and, less frequently, impersonal predicates and other complements: STALO OCEVIDNO ('it became apparent'). Only the copula BYT6 ('to be') can freely combine with all of the complement types.

The above discussion presents a summary of various areas which will have to be examined preparatory to undertaking a verb classification. Although a comprehensive classification of verbs may be a long way off, as suggested in the case of noun classification, certain of the features that can be expected to enter into any such scheme -- among them items discussed here -- can profitably be studied individually.

2.2.3 Adverbs

Based on materials contained in Prokopovich (1962), an effort was made to classify adverbial entries in the Russian Master Dictionary (RMD) according to their ability to combine with verbs, nouns, and adjectives. Two factors, however, led to the abandonment of this study as impractical: first, the adverbial entries in the RMD -- class (R7)N -- were found to include much linguistically unrelated material (words and phrases coded as "adverbs" in order to preclude their analysis in RMD applications); second, the bulk of the true adverbs were not in (R7)N at all, since they were handled in the RMD as derived from adjectival stems or by means of other routines.

The need for such a classification of adverbs is pressing, however, because little information is available in Soviet grammars. Some comments about verb-adverb constructions appear in the immediately preceding discussion of verb classification. Adverb-noun and adverb-adjective constructions were not studied in any degree of detail.

2. 3 Related language processing activities

As part of the statistically-oriented side of the grammatical research carried out during the contract period, the RAND corpus of Air Force Russian texts was partially processed for the purpose of obtaining statistical information on lexical frequency -- information which could serve both to measure the coverage of the Russian Master Dictionary (RMD) and as a guide to future lexical research. In addition, five updates of the RMD were carried out in order to incorporate the results of the work performed by the Library of Congress lexicographers.

Processing of the RAND Russian text corpus

About five million words of Russian text transcribed onto paper tape by the Air Force and later transferred to magnetic tape by the RAND Corporation were partially processed during the contract period with a view towards obtaining paradigm frequency statistics. A large part of the considerable amount of programming and processing involved in such an undertaking was completed, but further work had to be stopped at the end of the contract period, short of producing final results.

Initial difficulties involved rather tedious problems of code conversion and text editing. Once the text was listed in human-readable form, it was found that some 30 per cent of it was unusable due to such factors as garbling and reversal of paper tapes, thus reducing the total corpus to some three and one-half million Russian word form occurrences. When sorted and counted, this corpus was found to contain about 100, 000 unique word forms.

The next objective was to map word forms automatically onto their respective RMD stems in order to obtain a first approximation to paradigm frequency statistics. Since no appropriate program existed for matching Russian text words against stems in the RMD on a general-purpose computer, a new dictionary lookup program had to be written. The preparation of this program involved considerable effort in construction of usable tables of Russian endings from the original grammatical classification charts employed in preparing the RMD entries. Difficulties were encountered in processing multiple-stem (compound) Russian words.

Preliminary tests indicated that about 95 per cent of the word form occurrences in the corpus could be matched against the RMD. The remaining

5 per cent represented either words not in the dictionary or misspelled words, which are rather numerous in this corpus. Some 25,000 dictionary stems were needed to match those word forms that could be matched at all. This is about a quarter of the total number of single-word stems in the dictionary (phrase entries were ignored in this test). Only about 40 per cent of the 25,000 stems were matched against words with total frequency of occurrence greater than one in the corpus. Unfortunately, time did not permit pursuit of the investigation beyond the obtainment of these somewhat suggestive, but necessarily inconclusive, results.

Russian Master Dictionary updates

The bulk of the improvements incorporated into the RMD by the Library of Congress personnel between November of 1964 and August of 1965, together with other changes, additions, and deletions subsequent to the latter date were processed during the contract period. The changes in the RMD between August 1965 and July 1966 are reflected in Table II-13.

Table II-13: Size of the Russian Master Dictionary

RMD Version	Total number of entries	Date
181	165,202	8.19.1965
182	unavailable	11.13.1965
183	161,189	1.20.1966
184	152,541	5.04.1966
185	135,518	7.01.1966
186	135,225	7.27.1966

The size of respective updates is shown in Table II-14.

Table II-14: Size of the Russian Master Dictionary Updates

RMD Version	Updated on	New Version	Total adds and deletes in update
181	11.13.1965	182	10,244
182	1.20.1966	183	26,318
183	5.04.1966	184	19,092
184	7.01.1966	185	17,023
185	7.26.1966	186	503
total adds and deletes in RMD updating			73,180

NOTES

1. Sentential punctuation was not studied in detail. In declarative sentences, this is usually the period at the end of the sentence. However, in certain instances not considered in the present study (direct speech, for example), punctuation may appear on both sides of the sentence.

2. For actual word forms, the traditional parts of speech were used. These classes are not discussed as a group in this report, but individual parts of speech are described where pertinent to the discussion of particular sets of rules.

3. In Russian, the function words are: conjunctions, prepositions, particles, and interjections; all other part-of-speech classes are lexical words and include: nouns, verbs, adjectives, numerals, adverbs, pronouns, and words of the category of state ("forms in '-O'"). For English, see,e.g., Roberts (1954; 15-24).

4. Appositive ties are illustrated by constructions discussed in Part A4 of Section 2.1.4. See also note 29.

5. The term slovosochentanie is used here in the sense defined in V. V. Vinogradov, ed., Grammatika russkogo iazyka (Grammar of the Russian Language) (Vol. II.1, pp. 10-62), henceforth referred to as the "Academy Grammar".

6. Such constituents are the subject of a monograph by Cheshko (1960), but were not investigated as part of the present study. Occasional examples encountered in the 160-sentence sample of Pravda editorials were allowed to be absorbed by constituents right-adjacent to them.

7. Coordinative and subordinative conjunctions are not specifically enumerated. See Part C of Section 2.1.4 and note 66 for coordinative conjunctions. For further comments about subordinative conjunctions see Part D of Section 2.1.4.

8. This term is used in the sense defined by Nida (1960; 59). The extent to which strings of cardinal numerals can be called accumulative is open to question. However, there are sufficient formal similarities to justify such usage in this report.

9. This restriction refers only to what, according to Gleason (1961; 132), would be called immediate constituents of such constructions.

10. This is a temporary restriction which was intended to limit the scope of initial investigations. Exclusion of strings of characters other than

Russian is intended to avoid temporarily the need to consider such items as formulas, equations, Latin names, etc., since these have to be treated as part of the overall problem of substantivization which could not be studied for lack of time.

11. The effects of exhaustive automatic application of essentially context-free phrase-structure grammar rules requiring that each structure recognized by the grammar be analyzed into two immediate constituents have been described by Robinson (1966). What she described as the problem of overstructuring of endocentric constructions and the "doubtful propriety of permitting more than one way of structuring" is illustrated for Russian by the example STARYE STENY GORODA ('old walls of the city').

12. Instances of the latter type are illustrated by the contrasting interpretations of the sentence $ON KUPIL DOM V $LONDONE ('He bought a house in London').

13. Despite the importance of word order in surface structure recognition, the available information on Russian is fragmentary and much emphasis is placed on semantic distinctions which would effectively require discourse analysis capabilities. Two recent monographs (Sirotinina (1966) and Schaller (1966)) based on an extensive manually processed corpus contain interesting insights. Attempts to provide a theory of word order are reflected in Kholodovich (1966). During 1966 a number of articles in <u>Russkii iazyk v/shkole</u> dealt with the "semantic parsing" (<u>aktual'noe chlenenie</u>) of the sentence proposed by Mathesius in 1947. Chapters on word order are given in all the standard references used in the present study (see note 58). Gorbachik (1964) contains an interesting summary of the conventional statements about word order in Russian grammar.

14. The example is borrowed from Gvozdev (1961; 17). Another illustration is given at the end of the discussion of subjectless predications in Part B of Section 2.1.4.

15. This problem is disregarded in subsequent illustrations given in this report. For instance, if some constituent B and constituent C, agreeing in <u>number only</u>, produce a constituent D, then the hypothetical rules would have to be of the form

(a) B NG/X-P + C NG/X-P = D NG/X

(b) B NG/P + C NG/P = D NG/P

Note that the value of the NG attribute in (a) is set to "any except plural" and in (b) it is only plural.

16. One of the few instances where gender distinctions are significant in the plural is in dealing with adjective-noun agreement affected by cardinal numerals, where it is important to know the gender of the noun (cf. Part A8 of Section 2. 1. 4). Other instances occur in cases of agreement between coordinated adjectives and nouns.

17. In part responsible for the choice were the subjective preferences of those involved with the development of RG2; it appeared that an initial swelling in the number of experimental rules was preferable to a great number of tags, a situation which created considerable difficulties in RG1. With the availability of ordering of subrules as an option, the desired "transparency" of rules can be accomplished more economically than could be done for RG2 rules without employing the device of constituent renaming.

18. Short-form adjectives and short-form participles (SF) can function only predicatively, and hence are not considered here.

19. In the course of linguistic research carried out under the present contract, a variety of adjectival strings have been identified. However, they are not discussed in this report because of space limitations. Many of these strings result from coordination and can have the type of structure described for such compounds (2. 1. 4C). Some of the strings of asyndetic form have a variety of meanings which are either rare in expository writing (cf., e.g., repetition for emphasis -- VYSOKI1, VYSOKI1 DOM ('a very tall house')) or cannot at present be recognized, e.g., NOVA1, LUCWA1 JIZN6 ('a new (i.e.,) better life').

20. Ordinal numerals, which some authors (e.g., Vinogradov (1947; 233-6)) consider "ordinal adjectives", are treated as a separate constituent class R. The decision to follow the prevailing viewpoint was motivated by peculiarities of ordinal numeral-cardinal numeral agreement and the use of ordinal numerals in fractions. Both topics were studied during the contract period, but are not further discussed in this report.

In addition to distinguishing between active and passive participles, special consideration should have been given to reflexive participial forms, because their syntactic function approaches that of passive participles although their morphological properties are those of active participles.

21. In oblique cases (all cases except nominative and accusative), all numerals, except SC/S1, SC/T, and SC/M when they act as nouns, are in the same case as the noun, which must be plural. SC/S1 numerals

agree in all cases in case, number and gender. SC/M and SC/T numerals govern the noun in the genitive plural irrespective of the case they themselves are in.

22. Since tag values cannot start with a numeral but must begin with an alphabetic character, the letter "S" was arbitrarily chosen here as a prefix to "1", "2", etc.

23. These statements apply to all cardinal numerals whether spelled out in full or represented by Arabic numerals.

24. The fact that these numerals are really nouns from a morphological standpoint is demonstrated by their number and gender distinctions.

25. In most modern Russian prose, denominations greater than a billion are seldom spelled out and are accordingly not considered here.

26. Such other purposes include employment in the rules necessary for adjective-noun agreement affected by numerals, a topic which is discussed briefly in A8 of the present section.

27. Detailed information about punctuation of such predications has been collected from various sources but is not included for reasons of space limitations.

28. The use of this term has been checked, among others, in Jespersen (1964, 1965) and Roberts (1954). So-called appositive adjectives (a man, lean and hungry, walked in) are usually referred to in Russian grammars as "detached" (obosoblennye) except for Unbegaun (1957; 300). The "appositive genitive" (the city of San Francisco) and "appositive clauses" introduced by subordinating conjunctions (This does not explain the fact that he knew where to find it) are not called appositive in Russian. The English examples used in these illustrations are from Roberts (1954; 467-8). The sense in which the term is used in this report generally follows Rudnev (1959; 30-43).

29. In addition to the close appositions described below, six other types of appositions were studied during the contract period, but are not discussed at this time. The six types are illustrated by the following examples:

(1) Hyphenated appositions: JEN5INA-VRAC ('a woman doctor')
(2) Appositions where the appositive is enclosed in quotation marks: GAZETA "$PRAVDA" ('the newspaper Pravda')
(3) Appositions where the appositive is enclosed within parentheses: NEGUS (KOROL6) ('Negus (a king)')

(4) Detached appositive constructions: $PRAVDA, ODNA IZ KRUPNE1WIX GAZET V $S$S$S$R, ... ('Pravda, one of the largest newspapers in the USSR, ... ')

(5) Detached appositive constructions where the appositive is introduced by a special conjunctive word: $TOLSTO1, KAK PISATEL6, GENIALEN ('Tolstoy as a writer is a genius')

(6) Miscellaneous mixed types as, for instance, LETCIK-ISPYTATEL6 $NEUDACNIKOV ('test pilot Neudachnikov').

30. Animateness is based on grammatical criteria. For further details, see the discussion of subclassification of nouns in Section 2. 2. 1.

31. "Humans", as distinguished from non-humans (cf. Greek alogos), are real or imaginary animate beings capable of speaking. To designate all animate non-humans as animals is obviously imprecise, but convenient for present purposes.

32. Epithet is used as an equivalent of the Russian prozvishche or prozvanie. Generally, an epithet differs from a name in its indeclinability.

33. This feature is intended to single out nouns which either identify classes of terrestial and celestial locations or names of such locations. Thus OZERO ('lake'), PLANETA ('planet'), $BA1KAL ('Baikal'), and some others are "geographic" nouns. The following nouns are not: KANAVA ('ditch'), DNO ('bottom'), POLE ('field').

34. Botanical nouns identify species and genera of plant life as well as names of their members. Latin names have not been considered.

35. "Nomenclature items" include various product designations and usually consist of an abbreviation and/or numerals. For instance, $D$T-54 ('DT-54' -- a diesel tractor). Agent 007 and the characters in Zamyatin's We notwithstanding, it seems unreasonable to introduce this distinction for animate nouns in Russian.

36. This restriction specifically refers to nouns which can appear as constituents in appositions discussed in this report.

37. "Test A", for adjective-relatedness, applies only to constituents of the types shown in lines 7 and 8 of Table II-6, which both have the property of being able to combine as C_1 constituents with C_2's which are "titles" (cf. line 9 of Table II-6). Adjective-related C_1's (line 7) must always agree in gender with their C_2 in such constructions.

38. The statement should read in full: "Is the noun morphologically or semantically related to an adjective?" For example STARIK ('old

man') - STARY1 ('old'), BOGAC ('rich person') - BOGATY1 ('rich')...

39. Genera-species distinctions implied in the tests are intended to distinguish nouns so related: RYBA ('fish') and AKULA ('shark'); QVETOK ('flower') and LILI4 ('lily'); GAZ ('gas') and BUTAN ('butane'), etc. Taking an extreme position, completely satisfactory recognition of appositions would require encyclopedic information and discourse analysis capability. Thus, in a sentence like $QAR6 $4MOMOTO PRIKAZAL KAZNIT6, two analyses are possible.

(a) Tsar Yamomoto ordered an execution.

(b) The Tsar ordered Yamomoto to be executed.

To avoid the former analysis, it would be necessary to appropriately tag the words QAR6 ('tsar') and $4MOMOTO ('Yamomoto') in order to prevent the combination 'Tsar Yamomoto' since of all the tsars none was named Yamomoto. This type of problem affects all appositions discussed in this report with various degrees of severity, especially in instances where genera-species distinctions are concerned.

40. Additional refinements may be required. The problem of how to treat foreign names, especially Chinese or Arabic, was not studied in detail.

41. The number of nouns which "usually" do not function as members of close appositions is large. However, the examples listed nearly exhaust the instances where such function is extremely unlikely.

42. The first name-patronymic construction can be in apposition to the last name when the last name has another appositive: INJENER $PETROV, $IVAN $IVANOVIC ('engineer Petrov, Ivan Ivanovich').

43. Any member of an apposition can be a compound constituent which can be formed according to models sketched in (2.1.4C). However, it is the peculiarity of this type of apposition that when several titles qualify a person "in different planes" (Bylinskii and Rozental' (1959; 30)) such titles can form an accumulative string. For instance, KOMSOMOLEQ INJENER $IVANOV ('komsomol member, engineer Ivanov'). See also note 65.

44. The apposition recognized by this rule has agreement peculiarities involving the gender of adjectives and, where applicable, of verbs. Several articles, of which Protchenko (1961) is a fair example, have appeared about this problem. Rozental' (1965; 243) suggests a set of rules which are sufficiently comprehensive to eliminate the need for further discussion of this topic here.

45. The two options are: (a) agreement in case between the common noun and the proper name (POD GORODOM $KALUGO1 'near the city of Kaluga') or (b) instances where the proper name remains in the nominative (NA OZERE $IL6MEN6 'on lake Il'men''). A summary of current usage is provided in Rozental' (1965; 253-256).

46. The practical difficulty pointed up by this example, which becomes especially pronounced in the A12 subgroup (constructions with nomenclature items), is that even though a detailed subclassification can help avoid certain difficulties, the size of the dictionary required may easily become prohibitive. If, however, proper names are not included in the dictionary on a large scale, a personal name and a geographical name alternative may have to be considered for every proper name not found in the dictionary. Taking $VIARDO ('Viardot') as a possible substitution for $VOLGA in our example, the following interpretations are possible.

(a) Beyond the mountain, the Viardot becomes cooler.

(b) Beyond Mt. Viardot it becomes cooler.

(c) Beyond Viardot's mountain it becomes cooler.

(d) Beyond the mountain, Viardot starts to feel chillier.

In (a) and (b), Viardot is treated as a geographical name; in (c) and (d) as a personal name.

47. See Bylinskii and Nikol'skii (1957; 60). Possibly this subgroup of appositions should be considered as part of the A12 subgroup (constructions with nomenclature items).

48. The plural can occur only when one of the constituents is a compound constituent -- a situation not considered here.

49. Nomenclature items are frequently given enclosed in quotation marks (see the type of apposition shown in the second example in note 29). Criteria defining each option are not easy to recognize mechanically and the two options should be considered possible. However, since the use of appositives enclosed in quotation marks is not discussed in this report, only the other option is considered.

50. Comments made in notes 39 and 46 apply. In technical texts, appositive relations frequently are formally indiscernable from those of government. For instance, in a description of the functioning of the PT-1 semiconductor triode, the following references were encountered: TRIOD PT-1 ('PT-1 triode') . . . NA BAZU PT-1 ('on the base (of) PT-1 (triode)') . . . NA KOLLEKTOR PT-1 ('on the collector (of) PT-1 (triode)'), and so on. Moreover, if PT-1 is tagged as the name

45. The two options are: (a) agreement in case between the common noun and the proper name (POD GORODOM $KALUGO1 'near the city of Kaluga') or (b) instances where the proper name remains in the nominative (NA OZERE $IL6MEN6 'on lake Il'men' '). A summary of current usage is provided in Rozental' (1965; 253-256).

46. The practical difficulty pointed up by this example, which becomes especially pronounced in the A12 subgroup (constructions with nomenclature items), is that even though a detailed subclassification can help avoid certain difficulties, the size of the dictionary required may easily become prohibitive. If, however, proper names are not included in the dictionary on a large scale, a personal name and a geographical name alternative may have to be considered for every proper name not found in the dictionary. Taking $VIARDO ('Viardot') as a possible substitution for $VOLGA in our example, the following interpretations are possible.

(a) Beyond the mountain, the Viardot becomes cooler.

(b) Beyond Mt. Viardot it becomes cooler.

(c) Beyond Viardot's mountain it becomes cooler.

(d) Beyond the mountain, Viardot starts to feel chillier.

In (a) and (b), Viardot is treated as a geographical name; in (c) and (d) as a personal name.

47. See Bylinskii and Nikol'skii (1957; 60). Possibly this subgroup of appositions should be considered as part of the A12 subgroup (constructions with nomenclature items).

48. The plural can occur only when one of the constituents is a compound constituent -- a situation not considered here.

49. Nomenclature items are frequently given enclosed in quotation marks (see the type of apposition shown in the second example in note 29). Criteria defining each option are not easy to recognize mechanically and the two options should be considered possible. However, since the use of appositives enclosed in quotation marks is not discussed in this report, only the other option is considered.

50. Comments made in notes 39 and 46 apply. In technical texts, appositive relations frequently are formally indiscernable from those of government. For instance, in a description of the functioning of the PT-1 semiconductor triode, the following references were encountered: TRIOD PT-1 ('PT-1 triode') . . . NA BAZU PT-1 ('on the base (of) PT-1 (triode)') . . . NA KOLLEKTOR PT-1 ('on the collector (of) PT-1 (triode)'), and so on. Moreover, if PT-1 is tagged as the name

of a triode, it is not impossible to anticipate a construction like the following: 3TOT TRIOD PT-1 ZAMENIT6 NE MOJET ('this triode cannot replace the PT-1 (triode)').

51. This restriction is introduced in order to avoid redundant analyses which would otherwise result.

52. Among the other constituents are such items as NPNP, produced by a subrule discussed in A9 of Section 2.1.4 and NPB (cf. subrule (60) in 2.1.4C).

53. The ability of nouns to be simultaneously modified by adjectives on both sides requires further study as part of the larger problem of modifier strings. As noted in 19, a variety of adjectival strings have been identified in the course of studies carried out during the contract period. It is important to distinguish (a) the two basic forms of composition of adjectival strings (those consisting only of adjectives versus mixed strings of the type OSOBA4, TEXNICESKOGO POR4DKA, PAUZA ('a special pause of technical nature')); (b) the relative position of such strings (pre- versus post-positional with respect to the noun); and (c) obligatory versus optional detachment. Much of this information has to be obtained from a direct study of source materials because the available references provide a spotty picture and linguistic intuitions are inadequate to anticipate all of the logically possible combinations.

54. Formally, agreement in number is affected. Some of the more obvious instances are suggested in Rozental' (1965; 247-52). Tentative recognition rules covering the various possibilities have been worked out. However, they could only be tested manually and are not discussed in this report.

55. The usage is not settled in Modern Russian. The information shown in Table II-9 is incomplete (instances involving adjectives of the type QELY1 ('whole') and numerals of the type POLTORA ('one and one-half') are not considered). Instances marked as non-standard (preposition of adjectives to numerals of the S1 type, e.g., twenty-one) are not uncommon. Cf., e.g., the problems with *pluralia tantum* nouns which cannot combine with S1, S2, or S3 (e.g., 21, 42, or 73) numerals requiring nouns in the singular (*SOROK DVOE NOJNIQ 'forty-two scissors') but do combine with all other numerals requiring (genitive) plural (SOROK P4T6 NOJNIQ 'forty-five scissors'). For additional references, see Suprun (1964; 81-8), Listvinov (1965; 168-70), Rozental' (1965; 243-6).

56. In addition to the Academy Grammar sections on word groups (Vol. II.1; 113-353), relevant sections on prepositions and preposition-noun

phrases in Peshkovskii (1956) and Vinogradov (1947) were used as source materials.

57. Predications are named according to the type of sentences they produce.

58. Standard references are the Academy Grammar (Vinogradov (1960)) and the Moscow State University Grammar (Galkina-Fedoruk (1964)). In addition, Gvozdev (1952), Rudnev (1963), Peshkovskii (1956), and Valgina <u>et al.</u> (1962), among others, were regularly consulted.

59. In addition, style manuals like Rozental' (1965), specialized studies, specifically those of Ebeling (1958) and Gil'chenok (1964), and numerous other sources have been studied. While there is a great deal of repetition of basic facts, useful insights can be gained from these sources regarding individual facets of the problem.

60. In order to cope effectively with the problem of subject-predicate agreement, it is necessary to consider additional features discussed in Section 2. 2. 2.

61. See Galkina-Fedoruk (1964); also Galkina-Fedoruk (1958).

62. One of the better discussions of this topic is found in Kolshanskii (1965; 180-5).

63. This was not consistently carried through all of the rules. The discrepancies were only partially corrected.

64. One of the better descriptions is to be found in Gvozdev (1952), later reworked in Gvozdev (1961). In many regards the use of coordinative compounds is affected by considerations of style and hence considered a borderline grammatical problem.

65. This distinction is elaborated in Bylinskii and Rozental' (1959; 30). The concept of <u>odnorodnost'</u> ('homogeneity, uniformity') as it applies to modifiers, for instance, is positionally affected. Hence in NOVATOR PROIZVODSTVA TOKAR6 TOVARI5 $BORISOV ('industrial innovator, lathe operator, comrade Borisov'), the preposed appositives are not "homogeneous". However, in postposition they are and, as evidenced by the punctuation, form a coordinative compound: TOVARI5 $BORISOV, NOVATOR PROIZVODSTVA, TOKAR6. Some interesting insights are also found in Golovin (1959).

66. A good semantic analysis of relations expressed by coordinative conjunctions is given in Gvozdev (1952), Bylinskii and Rozental' (1959) and Figurovskii (1961; 12-16). Materials from these and other sources

(for instance, Shapkin (1964)) were collected but are not included in this report for reasons of space limitations.

67. Such expressive usage is not considered here, although the provision for this possibility exists in subrule (64). Conjunctions like "I" ('and') should be considered as having two alternatives: (a) conjunctions used once and (b) iterative conjunctions.

68. Paired conjunction constructs should be identified as being the first and the second part of a compound. This distinction is not expressed in the simplified rules that follow. The provisions necessary to ensure appropriate lexical correspondence among members of paired, iterative, and repeated single and paired conjunctions have also been omitted from these rules.

69. Typical instances are given in the official rules of Russian orthography and punctuation (Dobromyslov (1957)). More difficult cases are dealt with in Bylinskii and Rozental' (1959) and in Shapiro (1966).

70. The punctuation of this particular sentence is open to question and it may be argued that in order to obtain a second interpretation a comma should follow MUJCINY. If this were true, then the interpretation corresponding to (a) would be 'Inhabitants of the city (men) and women . . . awaited. . . '.

71. The section on subordinate clauses (Vol. II. 2; 266-380) was used in conjunction with the discussion of subordinate conjunctions in Vol. I.

72. Cf. International Business Machines Corporation (1964).

73. Although the examples may have additional meanings, only those specified should be considered.

74. Used here in the sense described, for instance, by Nida (1960; 45-8).

75. For additional discussion, see Vinogradov (1960), especially p. 239 in Vol. II. 1.

76. See Maslov (1965), especially pp. 70-79, for a brief survey of earlier work and an illustration of twenty-five Aktionsart distinctions currently recognized.

77. For further details, see appropriate sections of the references given in note 58.

REFERENCES

Andreyewsky, A. (1965) "Subclassification of Parts of Speech in Russian: Verbs". Proceedings of the 1965 International Conference on Computational Linguistics. New York.

Bylinskii, K. I., and Nikol'skii, N. N. (1957) Spravochnik po orfografii i punktuatsii dlia rabotnikov pechati (Orthography and Punctuation Manual for Workers of the Publishing Industry). Moscow: Izdatel'stvo "Iskusstvo".

____, and Rozental', D. E. (1959) Trudnye sluchai punktuatsii (Difficult Instances of Punctuation). Moscow: Izdatel'stvo "Iskusstvo".

Casey, R., and Nagy, G. (1966) "Recognition of printed Chinese characters", IEEE Transactions on Electronic Computers, Vol. EC-15, no. 1, February, pp. 91-101.

Cheshko, L. A. (1960) Izuchenie slov, grammaticheski ne sviazannykh s predlozheniem (The Study of Words Grammatically Unrelated to the Sentence). Moscow: Uchpedgiz.

Demidova, A. K., Motovilova, O. G., Shevchenko, G. D., Chaplygin, E. P. (1963) Naibolee upotrebitel'nye glagoly sovremennogo russkogo iazyka (The Most Used Verbs in Modern Russian). Moscow: Izdatel'stvo ANSSSR.

Dobromyslov, V. A., and Rozental', D. E. (1955) Trudnye voprosy grammatiki i pravopisaniia (Difficult Points of Grammar and Orthography). Moscow: Uchpedgiz.

____, and Rozental', D. E. (1960) Trudnye voprosy grammatiki i pravopisaniia (Difficult Points of Grammar and Orthography). Moscow: Uchpedgiz.

____, et al. (eds.) (1957) Pravila russkoi orfografii i punktuatsii (Rules of Russian Orthography and Punctuation). Moscow: Uchpedgiz.

Eaton, H. S. (1961) An English-French-German-Spanish Word Frequency Dictionary. New York: Dover Publications, Inc.

Ebeling, C. L. (1958) Subject and Predicate, Especially in Russian. (Preprint of Dutch contributions to the Fourth International Congress of Slavicists). 's-Gravenhage: Mouton & Co.

Figurovskii, I. A. (1961) Sintaksis tselogo teksta i ucheni cheskie pis'mennye raboty (Syntax of Discourse and Student Compositions). Moscow:

Uchpedgiz.

Galkina-Fedoruk, E. M. (1958) *Bezlichnye predlozhenia v sovremennom russkom iazyke* (Impersonal Sentences in Modern Russian). Moscow: Izdatel'stvo MGU.

____ (ed.) (1964) *Sovremennyi russkii iazyk* (Modern Russian). Chast' II *Morfologiia, Sintaksis* (Part II: Morphology and Syntax). Moscow: Izdatel'stvo MGU.

Gil'chenok, T. E. (1964) "O grammaticheskikh otnosheniiakh mezhdu podlezhashchim i skazuemym v sovremennom russkom iazyke" (On Grammatical Relationships Between Subjects and Predicates in Modern Russian). *Filologicheskie nauki*, no. 2, pp. 60-67.

Gleason, H. A. (1961) *An Introduction to Descriptive Linguistics.* New York: Rinehart and Winston.

Golovin, B. N. (1959) "K voprosu o razgranichenii odnorodnykh i neodnorodnykh opredelenii" (On the Problem of the Distinction Between Coordinative and Non-coordinative Modifiers). *Russkii iazyk v shkole*, no. 2 pp. 74-77.

Gorbachik, A. L. (1964) "Ob izuchenii poriadka slov v russkom iazyke" (On Studying the Word Order in Russian). In *Iz opyta prepodavaniia russkogo iazyka inostrantsam* (From the Experience of Teaching Russian to Foreigners). Moscow: Izdatel'stvo MGU.

Gvozdev, A. N. (1952) *Ocherki po stilistike russkogo iazyka* (Essays on Russian Stylistics). Moscow: Izdatel'stvo APNRSFSR.

____ (1961) *Sovremennyi russkii literaturnyi iazyk* (Modern Literary Russian). Chast' II: *Sintaksis* (Part II: Syntax). Moscow: Uchpedgiz.

International Business Machines Corporation. (1964) *Automatic Language Translation.* Final Technical Documentary Report (January 1, 1964 through December 31, 1964); Contract AF 30(602)-3301. Yorktown Heights, New York.

Jespersen, O. (1964) *Essentials of English Grammar.* University, Alabama: University of Alabama Press.

____ (1965) *The Philosophy of Grammar.* New York: W. W. Norton and Company.

Kholodovich, A. A. (1961) "K voprosu o gruppirovkakh slov v predlozhenii"

(Concerning the Problem of Word Groupings in the Sentence). In Academician I. I. Meshchaninov Festschrift *Problemy iazykoznaniia* (Problems of Linguistics). Leningrad: Izdatel'stvo LGU, pp. 223-243.

____ (1966) "K tipologii poriadka slov" (Toward a Typology of Word Order). *Filologicheskie nauki*, no. 3, pp. 3-13.

Kolshanskii, G. V. (1965) *Logika i struktura iazyka* (Logic and Language Structure). Moscow: Izdatel'stvo "Vysshaia Shkola".

Listvinov, N. G. (1965) *Voprosy stilistiki russkogo iazyka* (Problems of Russian Stylistics). Moscow: Izdatel'stvo "Mysl'".

Maslov, Iu. S. (1965) "Sistema osnovnykh poniatii i terminov slavianskoi aspektologii" (The System of Basic Concepts and Terms in Slavic Aspectology). In Maslov, Iu. S., and Fedorov, A. V. (eds.), *Voprosy obshchego iazykoznaniia* (Problems of General Linguistics). Leningrad: Izdatel'stvo LGU, pp. 53-80.

Nida, E. A. (1960) *A Synopsis of English Syntax.* In Elson, B. (ed.), *Linguistics Series No. 4.* Norman: Summer Institute of Linguistics of the University of Oklahoma.

Peshkovskii, A. M. (1956) *Russkii sintaksis v nauchnom osveshchenii* (Russian Syntax from a Scientific Point of View). Moscow: Uchpedgiz, 7th edition.

Popov, A. S. (1964) "Imenitel'nyi temy i drugie segmentirovannye konstruktsii v sovremennom russkom iazyke" (Nominative of Representation and Other Segmented Constructions in Modern Russian). In Muchnik, I. P., and Panov, M. V. (eds.), *Razvitie grammatiki i leksiki sovremennogo russkogo iazyka* (Evolution of Grammar and Vocabulary of Modern Russian). Moscow: Izdatel'stvo "Nauka".

Prokopovich, N. N. (1962) *Sochetaniia narechii s imenami prilagatel'nymi v sovremennom russkom iazyke* (Adverb-Adjective Constructions in Modern Russian). Moscow: Uchpedgiz.

Protchenko, I. F. (1961) "Formy glagola i prilagatel'nogo v sochetanii s nazvaniiami lits zhenskogo pola" (Verb and Adjective Forms in Constructions with [Masculine] Names Denoting Female Persons). In *Voprosy kul'tury rechi* (Problems of Good Usage in Speech), no. 3. Moscow: Izdatel'stvo ANSSSR, pp. 116-126.

Roberts, P. (1954) *Understanding Grammar.* New York: Harper and Brothers.

Robinson, J. (1966) "Endocentric constructions and the Cocke parsing logic". *Mechanical Translation*, Vol. 9, no. 1, March.

Rudnev, A. G. (1959) *Sintaksis oslozhnennogo predlozheniia* (Syntax of the Complicated Sentence). Moscow: Uchpedgiz.

____ (1963) *Sintaksis sovremennogo russkogo iazyka* (Syntax of Modern Russian). Moscow: Izdatel'stvo "Vysshaia shkola".

Schaller, W.H. (1966) *Die Wortstellung im Russischen* (The Word Order in Russian). München: Verlag Otto Sagner.

Shapiro, A. B. (1966) *Sovremennyi russkii iazyk: Punktuatsiia* (Modern Russian: Punctuation). Moscow: Izdatel'stvo "Prosveshchenie".

Shapkin, V. I. (1964) *Izuchenie soiuzov na urokakh russkogo iazyka* (The Study of Conjunctions in Russian Language Instruction). Moscow: Izdatel'stvo "Prosveshchenie".

Sirotinina, O. B. (1966) *Poriadok slov v russkom iazyke* (Word Order in Russian). Saratov: Izdatel'stvo SGU.

Suprun, A. E. (1964) *Imia chislitel'noe i ego izuchenie v shkole* (Numerals and Their Study in the School). Moscow: Uchpedgiz.

Unbegaun, B. O. (1959) *Russian Grammar*. London: Oxford University Press.

Valgina, N. S., Rozental', D. E., Fomina, M. I., and Tsapukevich, V. V. (1962) *Sovremennyi russkii iazyk* (Modern Russian). Moscow: Izdatel'stvo "Vysshaia shkola".

Vinogradov, V. V. (1947) *Russkii iazyk* (Russian Language). Leningrad: Uchpedgiz.

____ (ed.). (1960) *Grammatika russkogo iazyka* (Grammar of Russian; two volumes in three parts). Moscow: Izdatel'stvo ANSSSR.

III. PREDICTIVE SYNTACTIC ANALYSIS OF RUSSIAN

Warren J. Plath

III. PREDICTIVE SYNTACTIC ANALYSIS OF RUSSIAN

3.0 Introduction

In parallel with the work on CSA described in Sections I and II of this report, a small study was conducted on predictive syntactic analysis of Russian. The principal accomplishments in this latter area were: (1) modification of the multiple-path predictive Russian Syntactic Analyzer to bring it into operational status at the IBM Research Computing Center (Section 3.1); (2) expansion and revision of the dictionary for the Analyzer with emphasis on the syntactic properties of high-frequency function words (Section 3.2); and (3) testing and evaluation of the performance of the Analyzer on a sample of several thousand words of modern Russian text (Section 3.3).

Additional activities undertaken in connection with this portion of the project included a study of some of the recent literature on transformational grammar of English -- in particular, Lakoff (1965) and Rosenbaum (1967) -- as a source of potential insights into problems of Russian-English structural transfer. Unfortunately, since the time and programming effort required to make the Analyzer fully operational turned out to be substantially greater than originally anticipated, the output of the Analyzer (the second major ingredient of the proposed structural transfer study) was not available until the end of the contract period. Accordingly, this portion of the work on predictive analysis did not progress sufficiently to yield reportable results.

3.1 Programming Activities

The present section summarizes programming activities carried out in support of predictive syntactic analysis of Russian. The central focus of these activities was the process of bringing the multiple-path predictive Russian Syntactic Analyzer -- an exhaustive sentence parsing system originally developed at Harvard (Plath, 1963) -- into operational status at the IBM Research Computing Center. In order to provide an appropriate framework for the discussion, it will be helpful first to consider briefly the organization of the system and the functions of its major components.

As can be observed from Figure III-1, within this parsing system the process of obtaining syntactic analyses for the sentences of a Russian text that has been transcribed onto punched cards requires the sequential execution of three programs: the PRE-ANALYZER, SETSEN, and SYNTAX. The PRE-ANALYZER begins by performing a variety of preliminary operations on the input text, including serialization, code conversion, and formatting. When these steps have been completed, the program proceeds to assign each word in the text a set of syntactic alternatives with associated English

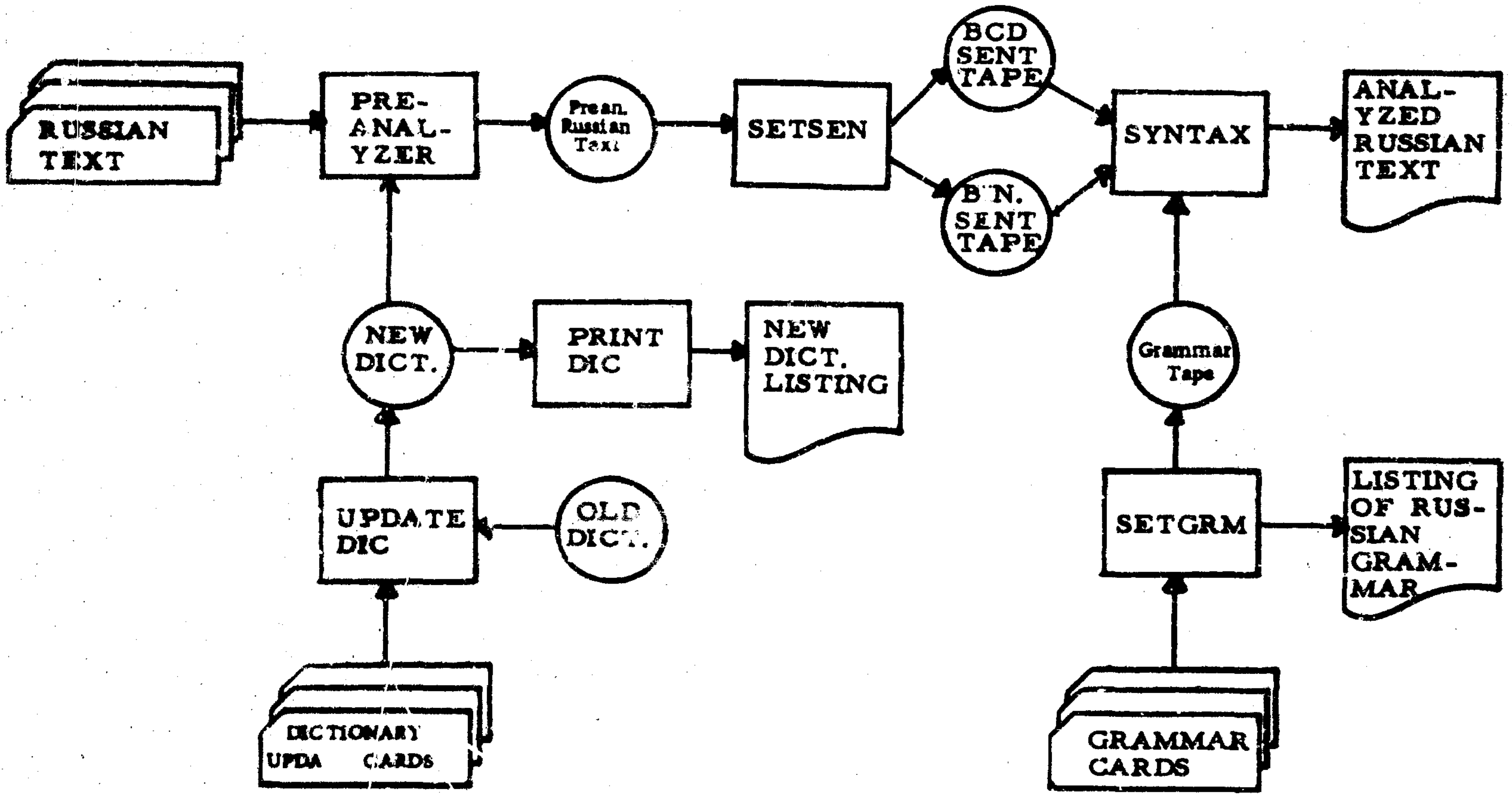


Figure III-1: SYSTEM ORGANIZATION OF THE RUSSIAN PREDICTIVE ANALYZER

correspondents through a process of dictionary lookup. The resultant output, known as the Preanalyzed Text Tape, serves as input to SETSEN. The primary function of the latter program is to convert BCD codes for syntactic alternatives into a considerably more compact binary representation compatible with that used on the Grammar Tape. In addition to a Binary Sentences Tape, which contains the compressed codes for the syntactic alternatives, SETSEN also produces a BCD Sentences Tape containing the original BCD codes together with their associated English correspondents. These two tapes, along with the aforementioned Grammar Tape, serve as inputs to SYNTAX, the predictive analysis program proper. SYNTAX systematically applies the rules of the Russian predictive grammar to each sentence on the Binary Sentences Tape according to the multiple-path predictive analysis algorithm whose details are presented in Plath (1963). The resultant output contains all surface structure analyses of each sentence that are consistent with the current grammar and the current dictionary. As each analysis is completed, SYNTAX edits it and writes it out in printable form, making use of the appropriate material from the BCD Sentences Tape.

In addition to the main sequence of programs just described, the predictive analysis system for Russian includes facilities for grammar and dictionary maintenance. Dictionary updating is performed by the UPDATE DIC program, which produces the dictionary tape file employed by the lookup phase of the PRE-ANALYZER. The tape file can be listed (with the Russian word forms transliterated) by using the PRINT DIC program. The grammar file, maintained in BCD form on cards, is converted into a more compact binary equivalent by the SETGRM program, which also produces a serialized BCD listing of the rules. Both the UPDATE DIC and the SETGRM programs include extensive provisions for the detection of format errors and invalid codes.

The first task performed in making the predictive analysis system operational at the IBM Research Computing Center was that of modifying all of its component programs to render them compatible with the local FMS II system. In most cases this was a relatively simple task, involving only minor modifications such as the alteration of tape assignments. For the programs with large storage requirements (the PRE-ANALYZER and SYNTAX), however, there were serious problems of storage overlap with FMS II system routines. These problems were eventually resolved with some difficulty at the price of doing without certain of the service programs normally available under the system.

Once the predictive analysis programs had been made compatible with the local FMS II system, it became possible to test them out on a large scale. While no difficulties were encountered in running SYNTAX and SETGRM, which had been extensively tested at Harvard over a considerable period of time, significant errors were discovered both in SETSEN and in the dictionary lookup routine of the PRE-ANALYZER, which had previously been checked out only on very limited samples of text. Beyond the modifications necessary to correct these errors, a number of other changes were made to the predictive analysis system in order to facilitate its employment for present research purposes. These latter changes included: (1) extension of the PRE-ANALYZER to permit acceptance of input text keypunched for the CSA system (Section I); (2) extension of the dictionary update package through development of a program for the automatic generation of update control cards for entries to be added to the dictionary file; and (3) improvement of the facilities for handling error conditions in SYNTAX.

In addition to the programming activites reported on above, which directly involved the predictive analysis system, a number of small support programs were written during the contract period to perform such varied functions as tape editing, dumping, and compilation of dictionary update statistics.

3. 2 Expansion and Revision of the Dictionary

During the contract period, the dictionary for the Russian predictive analyzer underwent considerable expansion, as well as a moderate amount of revision. The expansion process had two main objectives: the first was to provide syntactic coding and English correspondents for all lexical items occurring in the 160-sentence sample of Pravda editorials (Section 2. 1), thereby making it possible to employ the sample in testing the grammatical coverage of the system as a whole; the second was to extend the coverage of the dictionary to include additional high-frequency function words which might be expected to occur in a wide variety of texts.

Fulfillment of the first objective turned out to be a major undertaking, involving consultation of a variety of dictionaries and grammars in order to determine the appropriate syntactic coding for each of over 1500 lexical items. Once this task had been completed, however, fulfillment of the second objective was relatively easy, in that less than sixty additional items had to be processed in order to account for all high-frequency function words appearing in either of the two sources employed for word frequency data: (1) Kozak's list of approximately 1000 high-frequency word forms in Russian physics (Kozak, 1962) and (2) the 1000 most frequent word forms in the Air Force corpus described in Section 2. 3. In the total expansion process, the dictionary grew from a 1401-entry file covering 940 distinct forms to a 3315-entry file covering 2508 distinct forms -- a total increase in coverage of 1568 word forms.

Most of the revisions to the dictionary were made following the first complete analysis run on the entire 160-sentence Pravda sample. The great majority of the modifications involved new entries whose syntactic coding had been found in the course of analysis to be either incorrect or incomplete. A handful of the original entries were modified for similar reasons, while a somewhat greater number of them (primarily those for numerals and quantifiers) had their sets of syntactic alternatives revised to correspond to changes adopted for new entries with similar syntactic properties.

3. 3 Performance of the Analyzer on the Test Sample

Two complete analysis runs were made on the 160-sentence test sample of Pravda editorials, the first before the revision of the dictionary, and the second immediately thereafter. On the first run, one or more analyses were obtained for only 69 of the 160 sentences; the rest had no analyses. Furthermore, of the 69 sentences for which some analysis was obtained, only 58 had an analysis which was judged to be completely correct.* On the second run, the results were somewhat better, owing to an improvement of the grammatical coding in the dictionary: at least one parsing was obtained for each of 94 sentences, 78 of which had an analysis that was considered completely correct.

The results of the second run are summarized in greater detail in Table III-1, which displays for each sentence its serial number, length in words, number of analysis segments, error type (if any), and running time in minutes. With regard to performance in terms of running time for SYNTAX, the entire sample (3230 words) took 248. 69 minutes to process, or slightly in excess of four hours. As can be seen, analysis of some of the longest sentences was extremely time-consuming, while processing of sentences of average length (twenty words) generally took only a few seconds. Had the seven sentences of length greater than 40 words (i.e., sentences 34, 42, 45, 63, 88, 91, and 158) been eliminated, the remainder of the text (2883 words) would have been parsed in the more reasonable span of 76. 06 minutes.

*The criteria employed in deciding what constituted a "completely correct" parsing are in part subjective and hence somewhat difficult to describe. The basic requirement was that the given analysis represents a surface structure corresponding to the "normal" interpretation of the sentence with regard to such matters as: (1) the selection of syntactic alternatives; (2) the identity of grammatical subjects, objects, and complements within clauses; (3) the correct correspondence of modifiers with heads, relative pronouns with antecedents, and members of coordinate constructions with one another; and (4) nesting of infinitives and clauses within other clauses. One major systematic weakness of many of the "correct" analyses obtained by the system is the treatment of most prepositional phrases and some adverbs as "floating structures", i.e., as structures whose syntactic relationship to the rest of the sentence is completely unspecified.

Table III-1: Summary of Predictive Analyzer Performance on the Test Sample

Sent. No.	Sent. Length	No. of Anal.	Error Type	Time (min.)
1	13	4		0.18
2	6	1		0.03
3	20	6		0.31
4	8	0	B,F	0.04
5	23	0	A,E	0.10
6	29	0	A,B,E	0.07
7	31	0	B	0.18
8	24	0	A,C	0.16
9	30	0	A	0.38
10	11	1		0.04
11	23	1		0.08
12	9	2		0.08
13	9	0	C	0.04
14	10	2		0.07
15	14	1		0.05
16	9	1		0.06
17	10	1		0.05
18	12	0	A	0.10
19	9	1		0.03
20	7	1		0.04
21	15	2		0.20
22	14	1		0.11
23	22	1		0.12
24	16	2		0.25
25	19	4		0.21
26	25	8		0.48
27	19	3		0.69
28	15	3		0.11
29	12	2		0.07
30	15	1*	A	0.14
31	23	4		0.29
32	28	4		1.94
33	22	1		0.14
34	42	0	A	6.38
35	28	0	A,B,C, D,E	0.12
36	27	0	A	1.86
37	34	0	A,F	0.64
38	17	0	C	0.39
39	9	1		0.04
40	39	4*	A	7.01
41	11	1		0.04
42	41	2		1.30
43	8	1		0.05
44	9	0	C	0.03
45	50	0	A	2.80
46	30	4		1.32
47	40	0	A	3.33
48	30	0	A	1.73
49	30	0	A	0.45
50	26	0	F	0.54
51	37	0	A,B	5.04
52	23	1*	A	0.26
53	17	0	A	0.13
54	27	4*	F	0.72
55	19	1*	C	0.46
56	6	1		0.03
57	16	8		0.29
58	15	1		0.05
59	13	1		0.05
60	14	1		0.11
61	19	1		0.40
62	22	13		1.56
63	50	9*	A	102.35
64	25	0	B	0.15
65	15	0	A	0.06
66	18	1		0.09
67	25	0	A	0.40
68	18	3		0.45
69	20	0	B,E	0.07
70	8	9		0.17
71	10	1		0.03
72	16	0	A	0.13
73	14	4		0.30
74	21	0	A	0.05
75	19	3		0.14
76	19	3		0.12
77	8	1		0.04
78	12	1		0.07
79	22	0	A,D	0.05
80	20	8*	A	0.40

Table III-1 (contd.)

Sent. No.	Sent. Length	No. of Anal.	Error Type	Time (min.)	Sent. No.	Sent. Length	No. of Anal.	Error Type	Time (min.)
81	13	1		0.06	121	15	0	C	0.23
82	15	2		0.08	122	15	1		0.17
83	12	4		0.09	123	7	0	D	0.04
84	23	0	A	0.13	124	21	0	F	0.63
85	21	0	F	0.21	125	14	1		0.05
86	37	1*	A	1.65	126	17	4		0.28
87	14	0	B	0.06	127	32	0	A	0.71
88	64	0	A, F	27.74	128	18	0	A	0.07
89	14	2		0.06	129	13	1		0.10
90	15	0	D	0.19	130	14	2		0.24
91	42	0	A	25.45	131	28	0	C	0.76
92	39	0	A	7.15	132	19	3		0.12
93	24	0	A, B	0.37	133	11	4		0.20
94	13	0	A	0.53	134	22	2		0.16
95	20	3*	B	2.32	135	22	0	A	0.23
96	13	1		0.06	136	14	0	F	0.11
97	8	1		0.04	137	15	0	A	0.19
98	10	0		0.05	138	30	16*	A	3.17
99	22	2		0.25	139	11	3		0.08
100	19	5*	F	0.27	140	27	0	D	2.34
101	15	1		0.07	141	19	5		0.20
102	18	0	A, F	0.14	142	23	14		1.04
103	39	0	A	1.13	143	16	0	A, B	0.04
104	5	1		0.02	144	9	1		0.06
105	12	0	B	0.06	145	25	0	F	0.10
106	7	3**	F	0.08	146	12	1		0.05
107	25	0	A, C	0.26	147	11	1		0.04
108	22	0	A, C	0.08	148	21	1		0.27
109	23	0	B, F	0.19	149	22	0	A, E	0.26
110	24	0	A, C	0.08	150	39	0	A, C	0.12
111	27	0	B	0.29	151	6	1		0.04
112	18	0	A	0.26	152	22	2*	A	0.26
113	30	0	A, D	0.26	153	26	7		0.28
114	9	3		0.07	154	21	12*	D, F	0.71
115	8	2		0.05	155	36	4*	A	5.30
116	15	3		0.08	156	33	0	A, F	0.65
117	11	3		0.14	157	22	0	A	0.08
118	26	0	F	0.13	158	58	0	A, B, E	6.61
119	10	1		0.05	159	28	1		0.55
120	10	1		0.07	160	23	6*	F	2.84
					Totals	3230	279		248.69

* No analysis judged completely correct

** Non-sentence

Aside from making fundamental improvements in the parser, perhaps along the lines proposed by Kuno (1965), the problem of running time could be substantially reduced either by imposition of an arbitrary upper limit on the length of sentences processed or by running a successor to the present program on an interactive system.

Although the performance of the Analyzer in terms of running time was generally consistent with that observed earlier for a sample of scientific text (Plath, 1963; Thorpe, 1964), there was a significant degradation in performance with regard to the percentage of sentences for which correct analyses were obtained. For the sample of scientific text, 66 of 73 sentences, or about 90 per cent, received a correct analysis; whereas, in the present test, the percentage of sentences with correct parsings was slightly below 50 per cent. The explanation for this large discrepancy clearly does not lie in the change of vocabulary brought about by the shift from scientific text to editorials, since deficiencies in grammatical coding for the new lexical items were largely eliminated in the dictionary revision carried out prior to the second run. Moreover, although the random method of sentence selection may have led to greater than average syntactic diversity in the resulting test sample, examination of the output indicates that this effect was not sufficiently pronounced to account for more than a small part of the observed discrepancy. Instead, as will become evident in the discussion of error types below, the principal effect of the shift from scientific text to editorials was the employment with great frequency of one particular construction not provided for in the Russian predictive grammar.

Except for a few instances of incorrect handling of some of the typographical features of the input text, each of the errors detected involved failure to provide coverage for one or another feature of the surface syntax of the sentence in question. In an attempt to focus on recurrent problems, the errors have been grouped together into the following six types: Type A - instances of asyndetic coordination; Type B - new case constructions; Type C - new patterns of word order or nesting; Type D - new agreement relationships; Type E - input errors; and Type F - miscellaneous.

Type A errors, involving instances of asyndetic coordination, were by far the most frequent for the Pravda sample. No less than 54 of the 82 sentences with no completely correct analysis (or about two-thirds of them) contained at least one coordination of this type -- a construction for which there is no provision in the predictive Russian grammar. Briefly, an asyndetic coordinative construction (cf. Section 2.1.4C) is a coordination in which the components are linked by punctuation (usually commas or semicolons), but where no conjunction is present. Thus we have, for example, in sentence 30 of the test sample: $UVELICENIE IX ZAGRUZKI SPOSOBSTVUET ROSTU PROIZVODITEL6NOSTI TRUDA, SNIJENIH SEBESTOIMOSTI PRODUKQII, UVELICENIH NAKOPLENI1. ('Increase of their load promotes growth of productivity of labor, reduction of the net cost of production, increase of accumulations.') In this instance, the second member of the coordinative construction, SNIJENIH ('reduction'), was incorrectly analyzed as an appositive to the first member, ROSTU ('growth'); while the third member, UVELICENIH ('increase'), was similarly misinterpreted as standing in apposition to SNIJENIH. There were more numerous instances, however, where no analysis was obtained at all, either because the components of an asyndetic compound noun phrase did not satisfy the agreement requirements for appositions, or because the asyndetic coordination in question involved components of other syntactic types, such as verbs or prepositional phrases.

While it would be a trivial matter to alter the predictive grammar to provide for the recognition of asyndetic coordinations, under present circumstances the resulting increase in both the number of spurious analyses and the associated processing time would probably assume disastrous proportions. The only recourse would appear to be that of waiting until such time as it is possible to formulate much more stringent restrictions on what constitutes a linguistically acceptable coordination or apposition. In the meantime, it is somewhat reassuring to note that scientific writers do not seem to favor the use of asyndetic coordination to the extent that Pravda editorialists do; in fact, in the sample of scientific text processed earlier (Plath, 1963), not a single instance was found.

Except for the miscellaneous category (Type F), various case constructions not provided for in the current Russian grammar represented the second most frequent error type, with 15 occurrences. The bulk of these errors (12 occurrences) involved either (a) loosely governed instrumentals and datives of reference, not predictable from the verb, or (b) uses of the genitive peculiar to date constructions -- e.g., from sentence 69, . . . MARTOVSKI1 I SENT4BR6SKI1 ($$1965 GODA) PLENUMY . . . ('. . . the March and September (of the year 1965) plenums. . . '). The remaining three instances involved: (a) employment of the nominative case as a "vocative" (sentence 105) -- $VDUMA1S4, TOVARI5, V QIFRY PRIVEDENNYE V SOOB5ENII $Q$S$U. ('Think, comrade, of the figures presented in the communication of the Central Statistical Bureau.'); (b) use of the nominative in a quoted title in apposition to a noun in another case (sentence 109) -- . . . PO GAZETE "$STAVROPOL6SKA4 PRAVDA". . . ('. . . according to the newspaper "Stavropol'skaia Pravda". . .'); and (c) use of the genitive for what would normally be the subject of a negated passive verb (sentence 111) -- . . . V R4DE MEST NE PRINIMAETS4 VSEX MER K . . . ('. . . in a number of places not all measures are being taken towards . . . ').

There are 12 instances of errors involving failure to account for various word order and nesting arrangements (Type C). Five of them involved constructions in which both members of a compound verb had a common object, e.g. (sentence 35), -- ... UTVERJDAHT I RAZVIVAHT LENINSKIE NORMY ... ('... affirm and develop Leninist standards ...'). While such constructions could be handled within the framework of the predictive grammar without any of the unpleasant consequences that would ensue upon the elimination of Type A errors, a substantial increase in the number of subrules for verbal constructions would be required for this purpose. The remaining Type C errors stem from various word order patterns not covered by the present grammar. The majority of them are probably "normal" enough to warrant inclusion of appropriate provisions for them in the grammar, and it appears that very few additional subrules would be required for this purpose.

Seven agreement errors (Type D) were detected in the sample, three of them involving conjoined adjectives of different number, and the remainder having to do with plural agreement of singular nouns which can be considered to denote entities consisting of a number of components, i.e.: CAST6 ('part' - sentence 90), BOL6WINSTVO('majority' - sentence 113), OSNOVA ('foundation' - sentence 123), and MASSA ('mass' - sentence 140). Input errors (Type E) involved the loss of colons, semicolons, or parentheses in six sentences. Finally, seventeen sentences had errors of miscellaneous types, ranging from failure to detect instances of substantivization and ellipsis to non-recognition of certain punctuation patterns and discontinuous idioms.

In addition to those inadequacies of grammatical coverage explicitly recorded in the form of error type notations in Table III-1, other shortcomings are implicit in the numerous instances where multiple analyses were obtained. The total of 279 analysis segments works out to an average of about three analysis segments per sentence if only the 94 sentences which actually were given analyses are considered. This average compares favorably with the figure of approximately four analysis segments per sentence for the sample of scientific text; the reduction is largely attributable to the adoption of some of the recommendations made in the earlier study for alleviation of the problem of multiple analyses (Plath, 1963: 4-134 ff.).

Despite the reduction in the number of incorrect analyses, the residue (an average of two incorrect analyses for each correct one) is still very large. Examination of the output for the test sample indicates that much of the difficulty with multiple analyses is traceable to the continuing lack of adequate linguistic definitions of what constitutes either an acceptable coordination or an acceptable apposition, a point already mentioned in the discussion of Type A errors. Moreover, as indicated in the same discussion, attempts to extend grammatical coverage in certain directions, such as that of handling asyndetic coordination (or of tying prepositional phrases to specific governors), would seriously increase the number of undesirable analyses, given the present status of linguistic description of Russian.

A review of the results of the present experiment suggests the following two observations. The first is that, assuming that the results obtained are not entirely unrepresentative of the current state of the art, automatic syntactic analysis of Russian sentences must still be regarded to be in its experimental stages, and hence not yet ready for employment in text processing applications. Although much work remains to be done on the computational side, particularly if a capability for transformational recognition is desired, it appears that the more fundamental current obstacles to practical text processing application stem from unsolved linguistic problems of the sort alluded to in the preceding discussion. The second observation is suggested in part by the sharp differences in syntactic pattern observed between the sample of Pravda editorials and the sample of scientific text. While the prospect of constructing a huge array of interlocking "microgrammars" in order to handle texts of various types is an extremely uninviting one, the possibility of constructing a more restrictive grammar adequate for a single specific field appears worthy of serious exploration.

REFERENCES

Kuno, S. (1965) "The Predictive Analyzer and a Path Elimination Technique", *Communications of the ACM*, Vol. 8, No. 7, pp. 453-462.

Lakoff, G. (1965) "On the Nature of Syntactic Irregularity", *Mathematical Linguistics and Automatic Translation*, Report No. NSF-16. Cambridge, Mass.: Computation Laboratory of Harvard University.

Plath, W. J. (1963) "Multiple-path Syntactic Analysis of Russian", *Mathematical Linguistics and Automatic Translation*, Report No. NSF-12. Cambridge, Mass.: Computation Laboratory of Harvard University.

Rosenbaum, P. S. (1967) *English Grammar II* (forthcoming IBM publication).

Thorpe, R. (1964) "Revision of the Russian Analysis Program and Comparison of Processing Times", *Mathematical Linguistics and Automatic Translation*, Report No. NSF-13, Section III. Cambridge, Mass.: Computation Laboratory of Harvard University.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
The IBM Corporation
Thomas J. Watson Research Center
Yorktown Heights, N.Y. 10598

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
n/a

3. REPORT TITLE
Syntactic Analysis of the Russian Sentence

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final May 1965 - May 1967

5. AUTHOR(S) (First name, middle initial, last name)
Plath, Dr. Warren J.; Andreyewsky, Alexander; Strom, Robert E.; Lippman, Erhard O.

6. REPORT DATE
October 1967

7a. TOTAL NO. OF PAGES
158

7b. NO. OF REFS
52

8a. CONTRACT OR GRANT NO.
AF30(602)-3782

b. PROJECT NO.
4599

c.

d. 62405454

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
RADC-TR-67-484

10. DISTRIBUTION STATEMENT
This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of RADC (EMII), GAFB NY 13440.

11. SUPPLEMENTARY NOTES
Computational linguistics in Russian-English machine translation R&D

12. SPONSORING MILITARY ACTIVITY
Rome Air Development Center (EMIIH)
Griffiss Air Force Base, New York 13440

13. ABSTRACT

The report describes results of a two year research effort in the field of automatic syntactic analysis of Russian within the framework of Russian-English machine translation R&D.

The primary object of study and investigation consisted in design and development of the combinatorial syntactic analysis system, accompanied by an extensive linguistic research on Russian grammar. A concomitant small scale research on multiple path predictive syntactic analysis of Russian was conducted in parallel as an extension of the research effort initiated at Harvard University with the NSF support. Performance of the predictive analyzer on the test corpus of 160 Russian sentences is described on pp III 5 - 11 of the report. It was the contractor's intention to merge both automatic sentence parsing systems in the subsequent stages of Russian-English machine translation R&D.

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LIN
	ROLE	WT	ROLE	WT	ROLE
Russian-English Machine Translation R&D					
Computational Linguistics Research					
Combinatorial Syntactic Analysis					
Predictive Syntactic Analysis					
Computer Programming					

UNCLASSIFIED
Security Classification